arithmetic subroutines

```
1479          E * E
1480 UBOL IN E(6)
1481          M[6] = E
1482      U,  JUMP(0)                    "MAAK OF SCHOON
1483          E = MS                     "NEEM LAATSTE COEFFICIENT
1484          E * M[6]                   "X ARGUMENT
1485          E + MS[ - 2]               "+ COEFFICIENT
1486      U,  GOTOR(MC[ - 1])            "IF OF THEN ERUIT
1487          $ - 2
1488          JUMP( - 5)
1489 UENTIER(1)
1490          E + 0, P
1491 UENTIER1(12)
1492      U,  $ = E, E
1493      Y,  GOTOR(MC[ - 1])                    "ARGUMENTAL GEHEEL
1494          E - MT[7], E                       "+ 0 < E < .5?
1495      Y,  JUMP(4)                            "DAN ENTIER = + 0
1496      U,  $ = E, E                           "E - 0.5 INTEGER?
1497      Y,  GOTOR(MC[ - 1])
1498          E + DRT[11]
1499          E - DRT[11], Z
1500      Y,  E = 0
1501          GOTOR(MC[ - 1])
1502          - 65535                            ",5
1503          + 1

1504 UCOS(2)
1505          A = 1                      "COS(X) = SIN(X + PI/2)
1506          JUMP(1)
1507 U$IN(58)
1508          A = 0
1509          E * MT[54]                 "VERVANG X DOOR X * 2/PI
1510          MC = E, P
1511      U,  $ = E, E                   "* GEHEEL (EXP ≥ 0)?
1512      N,  E + DRT[11]                "ROND ANDERS X AF OP GEHEEL!
1513      N,  E - DRT[11], P
1514          A + G, E                   "BIJ COS: N = N + 1
1515      N,  A '*' 3, P                 "UITKIJKEN VOOR OVERFLOW IN A
1516          RCA(1), E                  "N EVEN
1517      Y,  $ = :MT[34]                "GEBRUIK DAN POLYNOOM VOOR SIN,
1518      N,  $ = :MT[21]                "GEBRUIK ANDERS POLYNOOM VOOR COS,
1519          E - MC[ - 2]               "MET ALS ARGUMENT Y = X - N, |Y|< 1)
1520      Y,  MC = - E                   "POLYNOOM VOOR SIN BEGINT MET Y
1521          E * E
1522          M[6] = E                   "BEREKEN
1523          E * MS[10]
1524          E + MS[8]                  "POLYNOOM
1525          E * M[6]
1526          E + MS[6]                  "VAN GRAAD 5
1527          E * M[6]
1528          E + MS[4]                  "IN Y - KWADRAAT
1529          E * M[6]
1530          E + MS[2]
1531          E * M[6]
1532          E + MS
1533      Y,  E * MC[ - 2]               "IN GEVAL VAN SIN, VERMENIGVULDIG
1534          LCA(1)                     "MET Y
1535      Y,  A + 1
1536          A '*' 2, Z                 "GEEN EXTRA MINTEKEN?
1537      N,  E = - E                    "KEER ANDERS TEKEN OM
```

```
1538          DO(MD[ - 1])
1539          GOTOR(MC[ - 1])
1540          + 0
1541          + 1                                   "C0 = 1
1542          + 1242657
1543          - 41522043                            "C2 = - , 123370055012₁₀+1
1544          - 1308641
1545          + 4067027                             "C4 = + , 253669507272₁₀+0
1546          + 1496389
1547          - 31298891                            "C6 = - , 208634689137₁₀-1
1548          - 1601776
1549          + 17393881                            "C8 = + , 919165417915₁₀-3
1550          + 1212415
1551          - 1708115                             "C10 = - , 248563446803₁₀-4
1552          - 1297852
1553          + 64660001                            "C1 = + , 157079632679₁₀+1
1554          + 1332904
1555          - 31928607                            "C3 = - , 645964097493₁₀+0
1556          - 1431346
1557          + 31668041                            "C5 = + , 796926261183₁₀-1
1558          + 1408717
1559          - 55253273                            "C7 = - , 468175259289₁₀-2
1560          - 1698552
1561          + 7584785                             "C9 = + , 160429269734₁₀-3
1562          + 1179647
1563          - 122197                              "C11 = - , 35564007032₁₀-5
1564          - 1333057
1565          + 25390670                            "2/PI = + , 636619772367₁₀+0

1566 UARCTAN(58)
1567               F + 0
1568               MC = G, R                        "BEWAAR OORSPRONKELIJKE TEKEN VAN X
1569          N,   F = - F                          "EN VERVANG X DOOR ABSX = ABS(X)
1570               MC = F                           "BERG ABSX OP
1571               S = 3                            "EN TEST
1572               F - 1, R                         "ABSX > 1?
1573          Y,   F = 1
1574          Y,   F/MC[ - 2]
1575          Y,   MC = F                           "VERVANG DAN X DOOR 1/ABSX
1576          N,   F = M[B - 2]                     "ZET ANDERS IN F ABSX TERUG
1577          N,   S - 2                            "EN ZET DE JUISTE INDICATIE
1578               F - MT[40], E                    "ABS(X) > 1 EN ABSX > TG(PI/12)?
1579          Y,   S - 1                            "GEEF JUISTE INDICATIE
1580          U,   A = 1, E                         "ABSX > TG(PI/12)
1581               F = M[B - 2]
1582          N,   JUMP(7)
1583               F * MT[37]                       "VERVANG
1584               F + 1                            "DAN
1585               MC = F                           "ABSX
1586               F = M[B - 4]                     "DOOR
1587               F - MT[33]                       "(ABSX - TG(PI/6))/
1588               F/MC[ - 2]                       "(1 + ABSX * TG(PI/6))
1589               M[B - 2] = F
1590               MC = S                           "INDICATIE OPBERGEN
1591               S = :MT[25]
1592               SUBC(:UPOL IN FF)                "BEREKEN POLYNOOM IN F KWADRAAT
1593               F * M[B + 1]                     "* ABSX = KWADRAAT
1594               F + 1
1595               F * MC[ - 3]                     "* ABSX
1596               S = MC[1]                        "INDICATIE
```

```
1597      U,  $ - 1, P                          "> 1?
1598      Y,  MC = - F
1599      N,  MC = F
1600          G = $, Z                          "INDICATIE = 0?
1601      N,  F * MT[21]
1602          F + MC[ - 2]
1603          $ = MC[ - 1], P                   "OORSPRONKELIJK TEKEN VAN X > 0?
1604      N,  F = - F
1605          DO(MD[ - 1])
1606          GOTOR(MC[ - 1])
1607          + 1365333
1608          + 22369812                        "- , 333333333246
1609          - 1369702
1610          + 40222387                        "+ , 199999980477
1611          + 1399661
1612          - 11933742                        "- , 142855496622
1613          - 1427237
1614          + 57172235                        "+ , 111044707738
1615          + 1403157
1616          - 59565289                        "- , 895216002193₁₀-1
1617          - 1458249
1618          + 43390644                        "+ , 622201788749₁₀-1
1619          - 1308524
1620          + 2669885                         "TG(PI/12) = + ,267949192430
1621          - 1305990
1622          + 43849281                        "TG(PI/6) = 1/SQRT(3) = 77350269190
1623          - 1334909
1624          + 43106668                        "PI/6 = + , 523598775599
1625  USQRT(47)
1626          F - 0, P                          "ARGUMENT < 0
1627      N,  F = 0                             "DAN RESULTAAT = + 0
1628      N,  GOTOR(MC[ - 1])
1629          MC = F                            "X BEHOUDEN
1630          A = F
1631          RUA(15)                           "ISOLEER EN
1632          MC = A                            "BEHOUD DE EXPONENT
1633          A = F
1634          $ = G
1635          F = :MC                           "BEHOUD STAPELWIJZER
1636          A '*' 32767                       "ISOLEER DE MANTISSE IN AS
1637          NORAS
1638          B = - :MC
1639          B - 26
1640          $ = B                             "BINAIRE EXPONENT + 2 IN $
1641          B = :MG                           "BRENG STAPELWIJZER TERUG
1642          $ + MC[ - 1], P                   "VORM COMPLETE BINAIRE EXPONENT + 2
1643      Y,  $ + 1
1644      U,  $ '*' 1, Z                        "WAS DEZE ONEVEN?
1645          $ '*' - 1
1646          MC = $
1647      Y,  RUA(1)
1648          MC = A
1649          RUA(3)                            "SCHAT SQRT(A):
1650          MC = A
1651          RUA(2)
1652          A * MT[19]
1653          M[B - 1] + A                      "X0 = ,3657 + (5/8) * A
1654          A = M[B - 2]
1655          RUA(4)
```

```
1656          DIVA(M[B - 1])                    "EERSTE NEWTON STAP:
1657          M[B - 1] + S                      "X1 = (A/X0 + X0)/2
1658          A = MC[ - 2]
1659          RUA(2)
1660          DIVA(MC)                          "TWEEDE NEWTON STAP:
1661          S + M[B + 1]                      "X2 = (A/X1 + X1)/2
1662          A = MC[ - 1]                      "NEEM BINAIRE EXPONENT
1663          LUAS(14), R                       "ZET 2 * X2 IN FLOATING OM
1664          E = MC[ - 2]                      "ZET X TERUG
1665          E/A
1666          A - MT[4], E
1667      N,  A - 32767                         "AS = X2/2
1668          E + A
1669          DO(MD[ - 1])
1670          GOTOR(MC[ - 1])
1671          '000 200 000'
1672          + 6135102                         ".09142

1673 ULN(77)
1674          E - 0, R                          "ARGUMENT > 0
1675      N,  E = - MT[72]                      "LEVER ANDERS - ONEINDIG AF
1676      N,  GOTOR(MC[ - 1])
1677          A = E
1678          RUA(15)                           "ZET DE EXPONENT APART EN
1679          MC = A                            "BEWAAR DIE
1680          A = E
1681          A '*' 32767                       "ZET IN PLAATS VAN DE
1682          A + MT[67]                        "BINAIRE EXPONENT 40
1683          S = G
1684          E = A
1685          E + 0                             "STANDAARDISEER DE EXPONENT
1686          A = E                             "OPNIEUW NE ZET DEZE WEER
1687          RUA(15)                           "APART
1688          M[B - 1] + A                      "TEL ERBIJ VORIGE EXPONENT
1689          A = E                             "VERVANG EXPONENT DOOR 0
1690          A '*' 32767
1691          S = G
1692          E = A                             "BEKIJK 40 BITS VAN E ALS FRACTIE
1693          LUAS(12)
1694          S = 0                             "ANALISEER DE EERSTE 27 BITS DAARVAN
1695          M[B] = A                          "OM DE FRACTIE
1696          RUA(3)                            "TE BRENGEN
1697          RLUSA(M[B]), R                    "IN HET INTEVAL
1698      Y,  S - 1                             "SQRT(8/9) < F < SQRT(9/8)
1699      Y,  JUMP( - 4)
1700          A = !MT[27]
1701          A - S
1702          G * MA                            "VERMENIGVULDIG FRACTIE MET
1703          MC = E                            "DE JUISTE MACHT VAN 9/8
1704          E + MT[41]                        "EN BERG NIEUWE FRACTIE OP
1705          MC = E
1706          E = MT[39]                        "BEREKEN
1707          E - MC[ - 4]                      "F1 = (1 - F)/(1 + F)
1708          E/MC                              "EN BERG
1709          MC = E                            "DEZE F1 OP,
1710          E * E
1711          MC = E
1712          E * MT[29]                        "BEREKEN
1713          E + MT[26]                        "POLYNOOM
1714          E * MC[ - 2]                      "IN
```

```
1715          E + MT[22]                          "F1 KWADRAAT
1716          E * MC[ - 2]
1717          MC = E
1718          G = $, Z
1719      N,  E * MT[16]                          "JUISTE VEELVOUD VAN LN(8/9)
1720          E + MC[ - 2]                        "OPTELLEN
1721          E + MT[12]                          "EN (1/2) * LN(9/8) ERBIJ OPTELLEN
1722          MC = - E
1723          G = MC[ - 3], Z                     "NEEM BINAIRE EXPONENT
1724      N,  E * MT[19]                          "* LN(2)
1725          E + MC[ - 1]
1726          DO(MD[ - 1])
1727          GOTOR(MC[ - 1])
1728          + 32768                             "2 ↑ 15
1729          + 36864                             "(9/8) * 2 ↑ 15
1730          + 41472                             "(9/8) ↑ 2 * 2 ↑ 15
1731          + 46656                             "(9/8) ↑ 3 * 2 ↑ 15
1732          + 52488                             "(9/8) ↑ 4 * 2 ↑ 15
1733          + 59049                             "(9/8) ↑ 5 * 2 ↑ 15
1734          - 1459121
1735          + 3895639                           "LN(9/8)/2 = +.588915178282₈-1
1736          +1426353
1737          -3895640                            "LN(8/9) = - .117783035656
1738          - 1269759
1739          + 6                                 "+.200000000002₈+1
1740          - 1332565
1741          + 44532834                          "+.666666478939
1742          - 1363134
1743          + 26681432                          "+.400433275889
1744          - 1332131
1745          + 35125202                          "LN(2) = +.693147180560
1746          + 506966                            "SQRT(8/9) * 2 ↑ 55 =
1747          + 65990648                          "+.339682755868₈+17
1748          '3777 37777'
1749          '3777 77777'                        "+.177664619751₈+629
1750          + 1310720                           "40 * D15

1751 UEXP(1)
1752          E + 0
1753 UEXP1(60)
1754          M[B] = E, P
1755      N,  E = - E                             "ABS(X)
1756          E - 1447, P                         "> 1447?
1757      N,  E = M[B]
1758      Y,  E = 1447                            "VERVANG DAN X
1759      Y,  S = - M[B + 1], P                   "DOOR 1447
1760      Y,  E = - 1447                          "MET CORRECT TEKEN
1761          E * MT[46]                          "*LOG(E) BEREKENING VAN 2 ↑ X
1762          MC = E, P
1763          SUBC(:UENTIER1)
1764          A = G                               "GEHELE DEEL VAN X
1765          E = MC[ - 2], Z                     "IS MANTISSE VAN X = 0?
1766      Y,  E = 1
1767      Y,  JUMP(10)
1768          MC = A
1769          E = - E                             "BEREKEN LOG(MANTISSE)
1770          E - MT[41], P
1771          E - MT[40]
1772      N,  E * MT[39]                          "BRENG ARGUMENT IN INTERVAL
1773          S = :MT[32]                         "- 1/2 < X < 0
```

```
1774          SUBC(:UROLINE)                    "EN BEREKEN DE POLYNOOM
1775     N,   E * E                             "IN(ARGUMENT) ↑ 2, ALS HET
1776          A = MC[ - 1]                      "ARGUMENT GEHALVEERD WAS
1777          A + 1                             "BRENG DAN GEHELE DEEL
1778     U,   A + 2048, R                       "IN ORDE
1779     U,   A - 2047, E                       "BUITEN INTERVAL( - 2047, 2047)
1780     N,   JUMR(5)
1781          A = A, R                          "> 2047
1782     N,   E/MT[27]
1783     N,   JUMR(6)
1784          E * MT[25]
1785          A - 2047
1786          LUA(15)
1787          A '*' - 32767
1788          S = 1
1789          E * A
1790          DO(MD[ - 1])
1791          GOTOR(MC[ - 1])
1792          - 1327104
1793          - 0                               "+.999999999999
1794          - 1332131
1795          + 35125162                        "+.693147180524
1796          - 1393280
1797          + 32498472                        "+.240226505508
1798          - 1460009
1799          + 4224865                         "+.555040853706₁₀-1
1800          - 1530010
1801          + 9820595                         "+.961794504001₁₀-2
1802          - 1627221
1803          + 23473550                        "+.133256313600₁₀-2
1804          - 1726494
1805          + 29964123                        "+.152132608000₁₀-3
1806          - 1735842
1807          + 34822787                        "+. 128376319999₁₀-4
1808          - 1298901
1809          + 37431644                        "LOG(E) = +.144269504089₁₀+1
1810          + 67076096
1811          + 1                               "2 ↑ 2047 = +. 161585030357₁₀+617
1812          - 65535
1813          + 1                               " .5

1814 UEXPNONNEG(3)
1815          E + 0, R                                "LT := BIT 41 VAN E
1816     U,   S = E, E                                "BIT 53 = LT?
1817          GOTO(MC[ - 1])

1818 UTTR(42)
1819          E + 0, Z                          "EXPONENT = 0?
1820     Y,   E = 1                             "ZO JA, RESULTAAT IS 1
1821     Y,   JUMR(21)
1822          MC = E
1823          E = M[B - 5]                      "NEEM GRONDTAL
1824          E + 0, Z                          "IS DAT 0?
1825     Y,   S = M[B - 1], R                   "EN EXPONENT > 0
1826     Y,   E = 0                             "DAN RESULTAAT IS 0
1827     Y,   JUMR(30)
1828          S = - M[B - 1], R
1829     Y,   S = - S
1830          A = - M[B - 2], E                 "EXP ≥ 0?
1831     N,   JUMR(7)
```

```
1832      U,  S + 2088, R                    "ABS(STAART VAN EXP) ≤ 2087
1833      Y,  A + 0, Z                       "^ KOP VAN EXP = 0?
1834      Y,  JUMR(10)
1835      U,  S = - G, R                     "GRONDTAL NEGATIEF?
1836      Y,  F = - F                        "DRAAI DAN TEKEN OM
1837      Y,  RUA(15), Z
1838      Y,  RCS(1), R
1839          SUBC(:ULN)                     "LN(GRONDTAL)
1840          F * MC[ - 2]                   "* EXPONENT
1841          SUBC(:UEXP1)                   "EN DIT ALS MACHT VAN E
1842      Y,  F = - F                        "NEEM ZONODIG DE INVERSE
1843          B - 2
1844          JUMR(14)
1845          F = 1, R
1846          M[B - 2] = F                   "BEGIN LOOP MET 1 EN CONDITIE YES
1847          F = M[B - 5]                   "GRONDTAL ↑ (2 ↑ I)
1848      N,  F * F                          "DAT WORDT(UITGEZONDERD DE EERSTE
1849      N,  M[B - 5] = F                   "KEER) GRONDTAL ↑ (2 ↑ (I + 1))
1850      U,  S '*' 1, Z                     "IS DEZE MACHT VAN HET GRONDTAL VAN BELANG?
1851      Y,  F * M[B - 2]                   "ZO JA, NEEM HEM DAN OP IN HET
1852      Y,  M[B - 2] = F                   "LOPEND PRODUKT
1853          RUS(1), Z                      "KLAAR?
1854      N,  JUMR( - 8)
1855          A + 0, R                       "WAS EXPONENT OORSPRONKELIJK NEGATIEF?
1856      Y,  F = 1
1857      Y,  F/M[B - 2]                     "NEEM DAN DE INVERSE
1858          B - 4
1859          DO(MD[ - 1])
1860          GOTOR(MC[1])

1861  UMOD(17)                               complex modulus
1862              M[B] = F, R
1863      N,      M[B] = - F
1864              F = M[B - 3] ,R
1865      N,      M[B - 3] = - F
1866      N,      F = - F
1867              M[B + 2] = F
1868              F - M[B] , R
1869              F = M[B]
1870      N,      M[B - 3] = F
1871      N,      F = M[B + 2]
1872              F + 0 , Z
1873      N,      F/M[B - 3]
1874      N,      F * F
1875              F + 1
1876      N,      SUBC(:USQRT)
1877              F * MC[ - 3]
1878              GOTOR(MC[1])

1879  UETTR(6)
1880              M[B + 1] = F
1881              S = M[B - 1]
1882              MC = S
1883              G = M[B -3]
1884              M[B - 3] = F
1885              F = M[B]
```

```
 0 JMTEREE(1)
 1          + 3                                  " MAGNETIC TAPES AVAILABLE
 2 UIETMT(10)
 3          :UARMT
 4          - 0
 5          - 0                                  " QUEUING REDS
 6          '760 071 000'(UNRMT)
 7          '660 071 000'(UNRMT)                 " IE := EALSE
 8          '760 072 000'(UNRMT)                 " LVIE := IRUE
 9          '660 072 000'(UNRMT)                 " LVIE := EALSE
10          '760 070 000'(UNRMT)                 " AE := IRUE
11          '200'
12          0                                    " POSITION IN FLIPFLOPWORD
13 JLABELMT(6)
14          0                                    " SLOTWOORD
15          0                                    " NEXT LABEL (IRRELEVANT)
16          0                                    " OPDRACHTWOORD
17          :JLABELMT(4)                         " VAKLABEL
18          0
19          0                                    " VAKSPECIFICATIE
20 JUNITLIST(4)
21          :JUNITADM1
22          :JUNITADM2
23          :JUNITADM3
24          - 0                                  " END MARKER, NEGATIVE

25 JTELBLOCKWRITE(3)
26          '470 000 411'                        " BROKWOORD
27          0
28          '470 000 411'                        " BROKWOORD
29 JTAREMWRITE(3)
30          '070 070 070'
31          '070 070 070'
32          '070 070 067'
33 JUNITADM1(8)
34          1                                    " UNIT NUMBER
35          - 1                                  " REEL NUMBER
36          0                                    " UNIT STATUS
37          - 0                                  " PRIVATE SV
38          - 0                                  " TAPE STATUS
39          - 0                                  " STORECODE
40          - 0                                  " TOTAL COUNT
41          - 0                                  " ERROR COUNT
42 JUNITADM2(8)
43          2
44          - 1
45          0
46          - 0
47          - 0
48          - 0
49          - 0
50          - 0
51 JUNITADM3(8)
52          3
53          200                                  " REELNUMBER OF SYSTEM TAPE
54          1
55          - 0
56          - 0
57          - 0
```

magnetic tape routines & variables permanently in core.

```
58          - 0
59          - 0
60  JMTEXSEM(3)
61          1
62          - 0
63          - 0

64  JUNITMES(46)
65          SUBC(:CREADDEC)
66          MC = $, Z
67      N,  $ - 3, E                        " - UNITNUMBER = 1, 2, OR 3 ?
68      Y,  GOTO(:CWRONG)
69          $ = :JUN:TMES[6]
70          GOTO(:CM:[30])
71          '031 400 000'[:JUNITMES + 8]    " IN
72          - '005 400 000'[:JUNITMES + 8]  " OUT
73          A = MC[ - 1]                    " RETRIEVE UNITNUMBER
74          $ + 0, Z                        " 0 FOR IN, 1 FOR OUT
75          A + :JUN:TLIST[ - 1]
76          A = MA                          " BASISADRES UNITADMINISTRATIE
77          E = :JMTEXSEM
78          UR
79          $ = MA[2]                       " UNIT STATUS
80      N,  JUMP(8)                         " OUT
81          $ - 4, Z
82      Y,  MA[2] = $
83      Y,  MA[1] = $
84      Y,  $ = MT[24]
85      Y,  A = 1
86      Y,  JMTEREE + A
87      N,  $ = MT[22]
88          JUMP(6)
89      U,  $ = - JMTEREE, P                " ANY MORE DISCONNECTIONS ALLOWED ?
90      N,  $ - 1, P                        " UNIT STATUS ≠ 0 OR 1
91      N,  JUMP(7)
92          $ - 3, Z
93      Y,  $ = MT[13]
94      N,  $ = MT[13]
95          E = :JMTEXSEM
96          UV
97          G = $
98          GOTO(:UENDMES)
99          $ = 4
100         MA[2] = $
101         $ =  - 1
102         MA[1] = $
103         JMTEREE + $
104         $ = MT[1]
105         JUMP( - 11)
106         JINVM1
107         CINVM9
108         JINVM2
109         JINVM3
110         CINVM13

111 CSPRONG(9)
112         $ = M[B - 2]
113         SUBCD(:UPROFANELINK[1])
114         A = M[B - 2]
115         A '*' - 511
```

message

```
116          A - MA
117          M[B - 2] - A
118          S = MC[ - 1]
119          G = MS
120          GOTO(:R46E3)
121 CMTRADRES(4)
122          R = :JMTEXSEM
123          UBROOD
124          G = JCURUNIT
125          GOTOR(MC[ - 1])
126 CMTVADRES(3)
127          R = :JMTEXSEM
128          UVWIT
129          GOTO(:CMTRADRES[2])

130 CREELCHECK(14)
131          A = G
132          S = :JUN:TLIST
133          G = MS, R
134     N,   GOTO(MC[ - 1])
135     U,   A - MG[1], Z
136     N,   S + 1
137     N,   JUMP( - 5)
138          A = MG[22]
139          A - :M0, Z
140          A = MG[1]
141     N,   JUMP( - 6)
142          A - 0, R
143          JCURUNIT = G
144          GOTO(MC[ - 1])
145 CREADBACK(6)
146          SUBC(:CREADKOR)
147          MINA(MG[16]), R                    " POS := POS - 2
148     Y,   A = - 1
149     Y,   GOTO(:CREADTAPE[9])
150          SUBCD(:CSPRONG)
151          CREADBACK1

152 CREADKOR(26)
153          SUBCD(:UNMORTAL)
154          SUBC(:CREELCHECK)
155     Y,   A = MG[2]
156     Y,   A - 2, Z
157     N,   A = 254
158     N,   JUMP(18)
159          A = MG[4]
160          A - 8, Z                           " TAPE STATUS = 8 ?
161     N,   A = 260
162     N,   JUMP(14)
163          A = MG[17], R                      " NEXTEL ?
164     N,   A = 261
165     N,   JUMP(11)
166          A = :MG[3]                         " :SV
167          LCA(9)
168          A + MG[16], Z
169          SUBCD(:PSE41)
170          S = MC[ - 1]                       " LINK
171          A = M[B - 1]
172          MC[ - 1] = R
173          M[B - 1] = A                       " INTERCHANGE OUTER LINK AND VALUE JUST READ
```

```
174           A = 2
175           G = JCURUNIT
176           GOTOR(S)
177           B = 1
178           GOTO(:CWRITETAPE[35])
179 CREADTAPE(14)
180           SUBC(:CREADKOR)
181           PLUSA(MG[16])                    " POS := POS + 2
182           S = MG[15]
183           LUS(1)
184       U,  A = S, R
185       N,  JUMP(2)
186           SUBCD(:CSPRONG)
187           CREADTAPE1
188           A = 1
189           MG[19] + A
190           SUBCD(:UMORTAL)
191           F = MC[ - 3]
192           DO(MD[ - 1])
193           GOTOR(MC[1])

194 CWRITETAPE(37)
195           A = MC[ - 2]
196           S = M[B]
197           M[B - 1] = S
198           MC = F
199           SUBCD(:UIMMORTAL)
200           SUBC(:CREELCHECK[1])
201       Y,  A = MG[2]
202       Y,  A - 2, Z
203       N,  A = 254
204       N,  JUMP(24)
205           A = MG[4]
206       U,  A - 4, R
207           A - 1, E                         " TAPE STATUS > 4 v TAPE STATUS = 1
208       Y,  A = 257
209       Y,  JUMP(19)
210           S = :MG[3]
211           LCS(9)
212           S + MG[16]
213           F = MC[ - 2]
214           MC = S
215           SUBCD(:RSE52)
216           G = JCURUNIT
217           A = 2
218           PLUSA(MG[16])
219           A - 510, R
220       N,  A = 1
221       N,  MG[19] + A
222       N,  A = 2
223       N,  MG[4] = A
224       N,  SUBCD(:UMORTAL)
225       N,  DO(MD[ - 1])
226       N,  GOTOR(MC[ - 1])
227           SUBCD(:CSPRONG)
228           CWRITETAPE1
229           B = 2
230           SUBCD(:UMORTAL)
231           GOTO(:UDYNERR)
232 CINSPTAPE(9)
```

```
233           SUBC(:CREELCHECK)
234       Y,  A = MG[2]
235       Y,  A - 2, Z
236       N,  A = 254
237       N,  GOTO(:UDYNERR)
238           G = MG[17]
239           S = G, R
240           DO(MD[ - 1])
241           GOTO(MC[ - 1])

242 CONSOLECLAIMED(1)
243           - 0
244 UDRUMTIME(6)
245           F = :USCS
246           S = M[220]
247           UP
248           S - M[220]
249           USTADRUM - S
250           GOTO(:UTEXTDRUM[2])
251 USTADRUM(1)
252           0

253 UGARR(8)
254           -0
255           -0                              "CASEDVTT
256           +0                              "UKLDV
257           -0                              "UKLACH
258           :UMAGAZYN[3]                    "UMAGADRES
259           36                              "UMAGAZYN, LETTERSHIFT, ROOD
260           2                               "CARRIAGE RETURN
261           8                               "NEW LINE

262 UMVR(4)
263           +0
264           +0                              "TTLABEL
265           +0
266           :UMAGAZYN[-1]

267 UIETTT(10)
268           :UARCPB
269           -0
270           -0
271           '760     071      023'          "IF (TT) := IBUE
272           '660     071      023'          "IF (TT) := EALSE
273           '760     072      023'          "LVIF (TT) := IBUE
274           '660     072      023'          "LVIF (TT) := EALSE
275           '760     070      023'          "AF (TT) := IBUE
276           0                               "1 IN FLIPFLOPWOORD
277           '100     000      000'          "VAN APPARAAT NO. 19

278 UIETRC(10)
279           :UARCKB
280           -0
281           -0
282           '760     071      024'          "IF := IBUE
283           '660     071      024'          "IF := EALSE
284           '760     072      024'          "LVIF := IBUE
285           '660     072      024'          "LVIF := EALSE
286           '760     070      024'          "AF := IBUE
287           0                               "1 IN FLIPFLOPWOORD VAN
```

```
288          '040     000      000'              "APPARAAT NO. 20

289 CASERC(1)
290          +0

291 UTWNR(1)
292          -0
293 EORSEM(3)
294          + 1
295          - 0
296          - 0

297                                                "COORDINATOR
298                                                "USW1
299 USW1(25)
300          SUBCD(:UREDDEN)
301          S = UWARNINGPOINT
302          S - GL24
303          S - USTEEK, P                         " P < WARNINGPOINT - STEEK?
304     Y,   JUMP(- 2)
305          S + USTEEK[USTEEK], P                 " P < WARNINGPOINT + STEEK?
306     Y,   S - USTEEK
307          S + GL24
308          UWARNINGPOINT = S                     " SET NEW WARNINGPOINT
309     N,   SUBCD(:UMES36)                        " SIGNAL DIVISION DRUM
310          JUMP(1)
311          SUBCD(:UREDDEN)                       " ENTRYPOINT UMOLYRMSEG
312          UTREXPROOD                            " RED P(UTREX)
313          S = UFIP, Z                           " FIP = 0?
314     Y,   GOTO(:ONZIN3)                         " DISASTER
315          S = 1
316          UFIP - S                              " DEC(FIP)
317          GL24 + S                              " INC(P)
318     U,   S = UFIOP, Z                          " FIOP = 0?
319     N,   UFIOP - S                             " DEC(FIOP)
320     Y,   S = 13
321     Y,   UAMVAR = S                            " AMVAR := 13
322          UTREXVWIT                             " WHITE V(UTREX)
323     Y,   SUBCD(:UROWNSEMADRES)                 " R(AMSEM)
324          GOTO(:UHERSTELTERUG)

325 UMOLYRMSEG(2)
326     U,   A = MS[0], Z
327     Y,   SUBCD(:USW1[11])
328 UCLEAR(4)
329          '377 737 777'
330          '377 737 777'
331          '377 737 777'
332          '377 737 777'
333 UIETTRA(10)
334          :UARTRA
335          - 0
336          - 0
337          '760 071 000'
338          '660 071 000'
339          '760 072 000'
340          '660 072 000'
341          '760 070 000'
342          '000 400 000'
343          0
```

strategic subroutine

```
344 UIETTRB(10)
345         :UARTRB
346         - 0
347         - 0
348         '760 071 000'[2]
349         '660 071 000'[2]
350         '760 072 000'[2]
351         '660 072 000'[2]
352         '760 070 000'[2]
353         '000 100 000'
354         0
355 UIETTRC(10)
356         :UARTRC
357         -0
358         -0
359         '760 071 000'[17]
360         '660 071 000'[17]
361         '760 072 000'[17]
362         '660 072 000'[17]
363         '760 070 000'[17]
364         '000 000 001'
365         0
366 USCHAKELTRA(6)
367         - 0
368         :USCHAKELTRA[4]
369         :UWIGTRA[- 1]
370         - 0
371         :USCHAKELTRA[1]
372         :UWAGTRA[- 1]
373 USCHAKELTRB(6)
374         - 0
375         :USCHAKELTRB[4]
376         :UWIGTRB[- 1]
377         - 0
378         :USCHAKELTRB[1]
379         :UWAGTRB[- 1]
380 USCHAKELTRC(6)
381         -0
382         :USCHAKELTRC[4]
383         :UWIGTRC[ - 1]
384         -0
385         :USCHAKELTRC[1]
386         :UWAGTRC[ - 1]
387 UBLANK(1)
388         UINVBLANK
389 UARROW1(1)
390         UINVARROW1
391 UARROW2(1)
392         UINVARROW2
393 USTMAXHERSEG(1)
394         UMAXHERSEG
395 USTHERPLENTY(1)
396         UHERPLENTY
397 USTHERREEL(1)
398         UHERREEL
399 UALLHOLE(1)
400         UINVALLHOLE
401 UALLHOLE4(1)
402         UINVALLHOLE4
```

```
403 CMNA21(1)
404         CINVM21

405 RSE0(21)                            " BLOCK ENTRY
406         G = MC[- 1]
407         SUBCD(:SPR)                 " STACK PAGE REQUEST
408         UWS[0] = G                  " SAVE LINK
409     N,  B = NSP                     " GLOBAL VARIABLE NEW STACK PAGE
410         F = MA[- 3]                 " PREPARE STANDARD LOCALS
411         MC[0] = F                   " CURRENT D AND SV LINK
412         MC[0] = A                   " RETURN LRP
413         A = :MC[0]                  " LRP IN A
414         S - 5, P                    " EXPLICIT LOCALS PRESENT?
415     Y,  F = UCLEAR                  " CLEARING CONSTANT
416     Y,  MC[2] = F                   " INITIALIZE TWO LOCALS?
417     Y,  S - 2, P                    " MORE LOCALS PRESENT?
418     Y,  JUMP(- 3)
419         B + :MS[2]                  " FIRST FREE PLACE IN B
420         MA[0] = B                   " SET WP
421         S = 1
422         PLUSS(MD[- 1])              " INCREASE CBM
423         MA[1] = S                   " STACK CBM
424         S + DPT[6]                  " CHANGE A = MD[CBM] IN MD[CBM] = A
425         DO(S)                       " FILL LRP IN DISPLAY
426         GOTOR(UWS[0])               " AND CONTINUE MAIN PROGRAM

427 RSE44(6)                            " PROCEDURE AS EXPRESSION
428         G = M[B - 1]                " MAKE SPACE FOR RESULT
429         M[B + 1] = G
430         F = UCLEAR                  " INITIALIZE RESULT
431         MC[- 1] = F
432         F = MS[2]                   " INVARIANT STARTING ADDRESS AND CONTEXT D
433         S = 9                       " 4 * NUMBER OF PARAMETERS + 9
434 RSE74(3)                            " FORMAL PROCEDURE ENTRY
435         B + 8                       " B ON TARGET AR
436         M[B - 8] = S                " FILL LRP[- 6]
437         GOTO(:R46E4)                " MERGE WITH NSIS - ENTRY

438 UTEXTDRUM(10)
439         GOTO(:UDRUMTIME)
440         UR                          " R(SCS)
441         SUBCD(:URIETDRUM)
442         S = M4[- 2]                 " :CTE
443         G = MS[0]                   " :SV
444         MG[0] = S                   " SV := :CTE
445         G = M4[- 8]
446         UV                          " V(USCAMSEM)
447         SUBCD(:URIETDRUM)
448         GOTO(:UTEXTDRUM)

449 R50E1(44)                           " MAKE PYSICAL ADDRESS
450         S = M[B - 3]                " INVARIANT ADDRESS IN S
451         RUS(9), P
452     N,  S '*' DPT[1]                " PRECAUTION MEMORY > 32K
453         A = MS[0], P                " SEGMENT ON CORE?
454     Y,  S = - MA[1], P              " ORIGINAL?
455     N,  JUMP(12)
456         S = 1
457         PLUSS(URAGECLOCK), Z        " INCREASE PAGECLOCK, OVERFLOW?
458     N,  S - MA[3], P                " PAGE PROFANE?
```

```
459     Y,  MA[3] + S, E                    " SET INTEREST, OVERFLOW ^ PROFANE?
460         S = MC[- 3]                     " INVARIANT ADDRESS
461     N,  S '*' 511, E                    " LINE NUMBER, OVERFLOW?
462     N,  S + MA[2]                       " FYSICAL ADDRESS
463     N,  DO(MD[- 1])                     " RESTORE LRR
464     N,  GOTOR(MC[0])                    " RETURN
465         B + 1
466         SUBCD(:UZEROHOUR)               " SCALE INTERESTNUMBERS
467         JUMP(- 12)
468         S = - S, R                      " COPY?
469     Y,  MA[1] - S                       " CTE[1] := - 0
470     Y,  SUBCD(:MEDR)                    " MAKE FREE DRUMPAGE
471     Y,  GOTO(:R50E1)
472         SUBCD(:UIMMORTAL)
473         A = A, Z                        " SV EMPTY?
474     Y,  SUBCD(:USW1)                    " SW1, P - INCREASE
475         S = - S
476         SUBCD(:SRQ)                     " SEGMENT REQUEST
477         A = MS[0]                       " :CTE[0]
478         S = DRT[2]
479         MA[3] - S                       " DECREASE HOLYNESS COUNTER
480     N,  JUMP(11)
481         MC[0] = E                       " SAVE E
482         S = B                           " SAVE B
483         B = MA[2]                       " :CRB[0]
484         E = UCLEAR                      " CLEARING CONSTANT
485         A = 512
486         MC[0] = E                       " INITIALIZE NEW PAGE
487         MC[0] = E
488         A - 4, Z
489     N,  JUMP(- 4)
490         B = S                           " RESTORE B
491         E = MC[- 2]                     " RESTORE E
492         SUBCD(:UMORTAL)
493         GOTO(:R50E1)
494 CRLUMES(20)
495         S + :ULISTRLU, Z
496         S = MS
497         UMIREAC[5] = S
498     N,  S = :CRLUMES[9]
499     Y,  S = :CRLUMES[17]
500         GOTO(:CMI[30])
501         '025 400 000'[:CRLUMES]         " ASP
502         '026 400 000'[:CRLUMES]         " B SP
503         - '034 400 000'[:CRLUMES]       " C SP
504         '033 400 000'[:UANS21]          " REELSS
505         '031 400 000'[:UMES27]          " INSS
506         - '005 400 000'[:UMES25]        " OUTSS
507         S + :ULISTRLU[UNUMTR], R
508         JUMP(- 13)
509         - '025 400 000'[:CRLUMES + 12]  " ASP
510         - 0                             " FOR LATER ADDITION
511         - '020 400 000'[:UANS16]        " OKSS
512         '004 400 000'[:UANSRL21]                "GOSS
513         '031 400 000'[:UMES27]                  "INSS
514         -'005 400 000'[:UMES25]                 "OUTSS

515 UHOLYCMSEG(7)
516         SUBCD(:SRQ)                     " SEGMENT REQUEST
517         SUBCD(:UREDDEN)                 " SAVE REGISTERS
```

request for segment holy in core

```
518          A = M$[0]                          " :CTE[0]
519          $ = MA[1],R                        " COPY?
520       Y, MA[1]-$                            " CTE[1]:=-0, ORIGINAL
521       Y, SUBCD(:MFDP)                       " MAKE FREE DRUM PAGE
522          GOTO(:UHERSTELTERUG)               " RESTORE REGISTERS AND RETURN

523 UTPVAR(11)                                  " REFERENCE POINT
524          - 0                                " UPUNFP
525          UNUMNORMALTP                       "UPUNACCEPT
526          UNUMNORMALTP                       "UPUNCASH
527          - 0                                " UPUNFREE
528          - 0                                " UPUNIDLE
529          :UTPSTRL1                          " BEGIN ADDRESS LIST PUNCH STREAMS
530          :UTPSTRL1[UNUMTPSTR]               " END ADDRESS LIST PUNCH STREAMS
531          :USBTPA                            " CYCLIC POINTER IN UPLUSEL
532          SUBCD(:UBEGINTPDOC)                " BEGIN PUNCH DOCUMENT
533          SUBCD(:UTPSEGMENT)                 " PROCES PUNCH SEGMENT
534          SUBCD(:UENDTPDOC)                  " END PUNCH DOCUMENT
535 UPLVAR(11)                                  " REFERENCE POINT
536          - 0                                " UPLOFP
537          UNUMPL                             " UPLOACCEPT
538          UNUMPL                             " UPLOCASH
539          - 0                                " UPLOFREE
540          - 0                                " UPLOIDLE
541          :UPLSTRL1                          " BEGIN ADDRESS LIST PLOTTER STREAMS
542          :UPLSTRL1[UNUMPLSTR]               " END ADDRESS LIST PLOTTER STREAMS
543          :USBPLA                            " CYCLIC POINTER IN UPLUSEL
544          SUBCD(:UBEGINPLDOC)                " BEGIN PLOTTER DOCUMENT
545          SUBCD(:UPLSEGMENT)                 " PROCES PLOTTER SEGMENT
546          SUBCD(:UENDPLDOC)                  " END PLOTTER DOCUMENT

547 UTPHANDLE11(18)
548          - 0                                " STREAMSEMAPHORE
549          - 0
550          - 0
551          - 0                                " AANHAAKLABEL
552          - 0                                " AFHAAKLABEL
553          :USBPM1['1 000 000']               " BOTTOM PM
554          - 0                                " DOCUMENT UNDER CONSTRUCTION
555          - 0                                " ATTACHMENT TIME
556          - 0
557          + 1                                " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
558          - 0                                " SV FOR PM SEGMENT
559          - 0                                " LUUFD
560          - 0                                " DESIRE
561          - 0                                " NFD
562          - 0                                " STREAM NOT SELECTED
563          :UTPVAR
564          :USBPM1[26]
565          - 0                                " BUILDING UP WORD

566 UTPHANDLE12(18)
567          - 0                                " STREAMSEMAPHORE
568          - 0
569          - 0
570          - 0                                " AANHAAKLABEL
571          - 0                                " AFHAAKLABEL
572          :USBPM1['2 000 000']               " BOTTOM PM
573          - 0                                " DOCUMENT UNDER CONSTRUCTION
574          - 0                                " ATTACHMENT TIME
```

```
575         - 0
576         + 1                                 " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
577         - 0                                 " SV FOR PM SEGMENT
578         - 0                                 " LUUFD
579         - 0                                 " DESIRE
580         - 0                                 " NFD
581         - 0                                 " STREAM NOT SELECTED
582         :UTPVAR
583         :USBPM1[26]
584         - 0                                 " BUILDING UP WORD

585 UTPHANDLE21(18)
586         - 0                                 " STREAMSEMAPHORE
587         - 0
588         - 0
589         - 0                                 " AANHAAKLABEL
590         - 0                                 " AFHAAKLABEL
591         :USBPM2['1 000 000']                " BOTTOM PM
592         - 0                                 " DOCUMENT UNDER CONSTRUCTION
593         - 0                                 " ATTACHMENT TIME
594         - 0
595         + 1                                 " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
596         - 0                                 " SV FOR PM SEGMENT
597         - 0                                 " LUUFD
598         - 0                                 " DESIRE
599         - 0                                 " NFD
600         - 0                                 " STREAM NOT SELECTED
601         :UTPVAR
602         :USBPM2[26]
603         - 0                                 " BUILDING UP WORD

604 UTPHANDLE22(18)
605         - 0                                 " STREAMSEMAPHORE
606         - 0
607         - 0
608         - 0                                 " AANHAAKLABEL
609         - 0                                 " AFHAAKLABEL
610         :USBPM2['2 000 000']                " BOTTOM PM
611         - 0                                 " DOCUMENT UNDER CONSTRUCTION
612         - 0                                 " ATTACHMENT TIME
613         - 0
614         + 1                                 " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
615         - 0                                 " SV FOR PM SEGMENT
616         - 0                                 " LUUFD
617         - 0                                 " DESIRE
618         - 0                                 " NFD
619         - 0                                 " STREAM NOT SELECTED
620         :UTPVAR
621         :USBPM2[26]
622         - 0                                 " BUILDING UP WORD

623 UTPHANDLE31(18)
624         - 0                                 " STREAMSEMAPHORE
625         - 0
626         - 0
627         - 0                                 " AANHAAKLABEL
628         - 0                                 " AFHAAKLABEL
629         :USBPM3['1 000 000']                " BOTTOM PM
630         - 0                                 " DOCUMENT UNDER CONSTRUCTION
631         - 0                                 " ATTACHMENT TIME
```

```
632             - 0
633             + 1                             " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
634             - 0                             " SV FOR PM SEGMENT
635             - 0                             " LUUFD
636             - 0                             " DESIRE
637             - 0                             " NFD
638             - 0                             " STREAM NOT SELECTED
639             :UTPVAR
640             :USBRM3[26]
641             - 0                             " BUILDING UP WORD

642 UTPHANDLE32(18)
643             - 0                             " STREAMSEMAPHORE
644             - 0
645             - 0
646             - 0                             " AANHAAKLABEL
647             - 0                             " AFHAAKLABEL
648             :USBRM3['2 000 000']            " BOTTOM PM
649             - 0                             " DOCUMENT UNDER CONSTRUCTION
650             - 0                             " ATTACHMENT TIME
651             - 0
652             + 1                             " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
653             - 0                             " SV FOR PM SEGMENT
654             - 0                             " LUUFD
655             - 0                             " DESIRE
656             - 0                             " NFD
657             - 0                             " STREAM NOT SELECTED
658             :UTPVAR
659             :USBRM3[26]
660             - 0                             " BUILDING UP WORD

661 UTPHANDLE41(18)
662             - 0                             " STREAMSEMAPHORE
663             - 0
664             - 0
665             - 0                             " AANHAAKLABEL
666             - 0                             " AFHAAKLABEL
667             :USBRM4['1 000 000']            " BOTTOM PM
668             - 0                             " DOCUMENT UNDER CONSTRUCTION
669             - 0                             " ATTACHMENT TIME
670             - 0
671             + 1                             " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
672             - 0                             " SV FOR PM SEGMENT
673             - 0                             " LUUFD
674             - 0                             " DESIRE
675             - 0                             " NFD
676             - 0                             " STREAM NOT SELECTED
677             :UTPVAR
678             :USBRM4[26]
679             - 0                             " BUILDING UP WORD

680 UTPHANDLE42(18)
681             - 0                             " STREAMSEMAPHORE
682             - 0
683             - 0
684             - 0                             " AANHAAKLABEL
685             - 0                             " AFHAAKLABEL
686             :USBRM4['2 000 000']            " BOTTOM PM
687             - 0                             " DOCUMENT UNDER CONSTRUCTION
688             - 0                             " ATTACHMENT TIME
```

```
689          - 0
690          + 1                                " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
691          - 0                                " SV FOR PM SEGMENT
692          - 0                                " LUUFD
693          - 0                                " DESIRE
694          - 0                                " NFD
695          - 0                                " STREAM NOT SELECTED
696          :UTRVAR
697          :USBRM4[26]
698          - 0                                " BUILDING UP WORD
699 UTRHANDLE51(18)
700          -0                                 "STREAMSEMAPHORE
701          -0
702          -0
703          -0                                 "AANHAAKLABEL
704          -0                                 "AFHAAKLABEL
705          :USBRM5['1 000 000']               "BOTTOM PM
706          -0                                 "DOCUMENT UNDER CONSTRUCTION
707          -0                                 "ATTACKMENTTIME
708          -0
709          +1                                 "DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
710          -0                                 "SV FOR PM SEGMENT
711          -0                                 "LUUFD
712          -0                                 "DESIRE
713          -0                                 "NFD
714          -0                                 "STREAM NOT SELECTED
715          :UTRVAR
716          :USBRM5[26]
717          -0                                 "BUILDING UP WORD
718 UTRHANDLE52(18)
719          -0                                 "STREAMSEMAPHORE
720          -0
721          -0
722          -0                                 "AANHAAKLABEL
723          -0                                 "AFHAAKLABEL
724          :USBRM5['2 000 000']               "BOTTOM PM
725          -0                                 "DOCUMENT UNDER CONSTRUCTION
726          -0                                 "ATTACKMENTTIME
727          -0
728          +1                                 "DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
729          -0                                 "LUUFD
730          -0                                 "DESIRE
731          -0                                 "NFD
732          -0                                 "STREAM NOT SELECTED
733          -0
734          :UTRVAR
735          :USBRM5[26]
736          -0                                 "BUILDING UP WORD

737 ULISTRLU(UNUMRLU)
738          :USBTRA
739          :USBTRB
740          :USBTRC
741          :USBRLA
742 RSE100(9)
743          S = MC[- 1]                  " LINK IN S
744          B = MA[0]                    " B := VP
745          F = MS[0]
746          MA[6] = F                    " FILL STATEMENT AND ELEMENT LINK
747          A = MA[3]                    " PARAMETERCODE CONTROLLED VARIABLE
```

```
748             A '*' - 259, Z                  " REAL OR INTEGER VARIABLE PURE?
749         Y,  GOTO(:R46E3[- 1])               " MERGE WITH NSIS ENTRY, JUMP TO ELEMENT LINK
750             A = 221                                 "ALARM CONTROLLED VARIABLE OF INCORRECT TYPE
751             GOTO(:UDYNERR[1])

752 RSE101(6)                                    " MAKE NEW ELEMENTLINK
753             S = M[B - 2]                    " LINK OBJECT TEXT
754             S '*' - 511                     " SUBTRACT LINE NUMBER
755             S - MS[0]                       " SUBTRACT INVARIANT ADDRESS 0-TH WORD
756             S - M[B - 2]                    " SUBTRACT LINK
757             MA[7] = - S                     " INVARIANT RETURN ADDRESS IN ELEMENTLINK
758             GOTOR(MC[- 1])                  " RETURN

759 RSE102(4)                                    " INITIALISATION STEP-UNTIL ELEMENT
760             SUBCD(:RSE101)                  " NEW ELEMENT LINK
761             MA[8] = - B                     " STEP := FALSE
762             S = MC[- 1]
763             GOTO(:MS[4])

764 RSE104(7)                                    " INCREMENTING
765             MA[11] = E                      " INCREMENT := E
766             S = MA[8], R                    " STEP TRUE?
767         Y,  E + MA[9]
768         Y,  GOTOR(MC[- 1])
769             MA[8] = B                       " STEP := TRUE
770             S = MC[- 1]
771             GOTO(:MS[2])

772 RSE103(3)                                    " TEST STEP-UNTIL ELEMENT
773             E - MA[9], Z                    " CV - NEWVALUE = 0?
774         N,  E = MA[11], E                   " INCREMENT SAME SIGNE?
775         N,  E + 0, Z                        " INCREMENT ZERO?
776 R103E1(3)                                    " ENTRY WHILE ELEMENT
777         Y,  JUMR(5)                         " JUMP TO STATEMENT
778             SUBCD(:RSE101)                  " NEW ELEMENT LINK
779             GOTOR(MC[- 1])
780 R103E2(2)                                    " JUMP TO NEXT ELEMENT
781             G = MA[7]                       " ELEMENT LINK
782             JUMR(2)
783 R103E3(4)                                    " ENTRY ARITHMETIC ELEMENT
784             SUBCD(:RSE101)                  " NEW ELEMENTLINK
785             G = MA[6]                       " STATEMENT LINK
786             S = MC[- 1]
787             GOTO(:R46E3[- 1])               " MERGE WITH NSIS-ENTRY

788 WFINDOCSEL(23)                               " FINISHED DOCUMENT SELECTION
789         U,  S = MG[4], R                    " EXIST CORRESPONDING IDLE PLUNCHER?
790         N,  GOTOR(MC[- 1])
791             SUBCD(:UREDDEN)                 " SETS M[B - 5] POSITIV, STREAMSELECTED := FALSE
792             S = MG[5]                       " :UTRSTRL1 OR :URLSTRL1
793             G = - UCON3                     " - (2 ↑ 26 - 1)
794             M[B] = E                        " MAX. STREAM LENGTH := - (2 ↑ 26 - 1)
795             A = MS[0]                       " :URLUHANDLE
796         U,  S = MA[13], R                   " NFD > 0?
797         N,  JUMR(7)
798         U,  S = - MA[14], R                 " NON SELECTED?
799             G = MA[0]                       " LENGTH IN PORTIONS
800         Y,  E - MA[7]                       " - ATTACHMENTTIME
801         Y,  E - M[B], R                     " LENGTH - ATT. TIME > MAXIMUM?
802         Y,  E + M[B]
```

Subroutines for evaluation of for list elements etc. etc.

```
803     Y,  M[B] = F                        " NEW MAXIMUM
804     Y,  N[B - 5] = - A                  " - :URLUHANDLE STREAM
805         S + 1
806         F = M[B - 2]                    " RETURN G UPON CALL
807     U,  S - MG[6], Z                    " READY?
808     N,  JUMP(- 14)
809         A = - M[B - 5], P               " STREAM CHOSEN?
810     Y,  SUBCD(:URLUSEL)                 " PLUNCHER SELECTION
811         GOTO(:UHERSTELTERUG)

812 UDISCRLU(21)                            " DISCONNECT PLUNCHER
813         SUBCD(:UREDDEN)
814         S = :ULISTRLU                   " :ULISTRLU
815         MC[0] = S
816         G = MS[0]                       " :USBRLU
817         S = 1
818     U,  S - UAMVAR[UG], Z               " PLUVAR = 1?
819     Y,  A = URLUDISC[UG], P             " DISC?
820         A = URLUTYPE[UG]                " :UTR(RL)VAR
821         MA[2] - S                       " DEC(PLUCASH)
822     Y,  SUBCD(:UBANKERSALLOWANCE)
823     N,  MA[2] + S                       " INC (PLUCASH)
824     Y,  MA[4] - S                       " DEC (PLUIDLE)
825     Y,  A = 3
826     Y,  UAMVAR[UG] = A                  " PLUVAR := 3
827     Y,  URLUDISC[UG] = - B              " DISC := EALSE    .
828     Y,  F = :UAMSEM[UG]
829     Y,  SUBCD(:UVOR[2])
830         S + MC[- 1]
831     U,  S - :ULISTRLU[UNUMRLU], Z       " ALL PLUNCHERS EXAMINED?
832     N,  JUMP(- 18)
833         GOTO(:UHERSTELTERUG)

834 UCASHINCREASE(6)                         " CASH INCREASE
835         S = MG[0], P                    " PUN(PLO) FP > 0?
836     Y,  UFPAS + S                       " INC(FPAS, PUN(PLO) FP)
837     Y,  MG[0] - S                       " PUN(PLO) FP := 0
838     Y,  SUBCD(:UFPASWAKING)             " USE FPAS FOR WAKING
839         SUBCD(:UDISCRLU)                " DISCONNECT PLUNCHER
840         GOTO(:UPROCBANKER)              " PROCEDURE BANKER, RETURN

841 UBANKERSALLOWANCE(32)
842         SUBCD(:UREDDEN)
843         UWORKB = B
844         G = UPLOCASH
845         UPLOFREE = G                    " FREE MONEY := CASH
846         G = UPUNCASH
847         UPUNFREE = G                    " FREE MONEY := CASH
848         B = :ULISTRM[UNUMRM + 1]
849         G = MC[0]                       " G = :USBRM
850         UFINISHDOUBTFUL[UG] = B         " FINISH DOUBTFUL := IBUE
851     U,  B - :ULISTRM[1], P
852     Y,  JUMP(- 4)
853         S = UNUMRM                      " N DOUBTFUL CUSTOMERS IN S
854         B = :ULISTRM[UNUMRM + 1]
855     U,  B - :ULISTRM[1], P
856     N,  JUMP(14)
857         G = MC[0]                       " G = :USBRM
858         A = - UFINISHDOUBTFUL[UG], P    " FINISH DOUBTFUL FALSE?
859     N,  A = UCLAIMRUN[UG]
```

```
860      N,  A - UBUNFREE, P                    " CLAIMPUN[I] > PUNFREE?
861      N,  A = UCLAIMPLO[UG]
862      N,  A - UPLOFREE, P                    " CLAIMPLO[I] > PLO FREE?
863      Y,  JUMP(- 9)
864          A = ULOANPUN[UG]
865          UBUNFREE + A                       " FREEPUN + LOANPUN[I]
866          A = ULOANPLO[UG]
867          UPLOFREE + A                       " FREEPLO + LOANPLO[I]
868          UFINISHDOUBTFUL[UG] = - B          " FINISHDOUBTFUL := FALSE
869          $ - 1, Z                           " DOUBTFUL CUSTOMERS = 0?
870      N,  JUMP(- 17)
871          B = UWORKB                         " RESTORE B
872          SUBCD(:UHERSTEL)
873          GOTO(MC[- 1])                      " RETURN

874 UINCFPASLUUFD(6)
875          G = MA[5]                          " G = :USBRMI
876          $ = MA[11]                         " $ := LUUFD
877          UFPAS + $                          " INC(FPAS, LUUFD)
878          MA[11] - $                         " LUUFD := 0
879          $ = MA[12], P                      " DESIRE > 0?
880      Y,  JUMP(8)
881 UFPASWAKING(13)
882          $ = UFPAS, P                       " FPAS > 0?
883      Y,  $ = 11
884      Y,  SUBCD(:UEXPMTRVAR)                 " EXIST(PMSHR = 11)?
885      N,  GOTOR(MC[- 1])                     " RETURN
886          A = UBLOCKSTR[UG]                  " :HANDLE BLOCKING STREAM
887          $ = 1
888          UFPAS - $                          " DEC(FPAS)
889          MA[11] + $                         " INC(LUUFD)
890          MINS(MA[12])                       " DESIRE := 0
891          UAMVAR[UG] = $                     " PMSHR := 0
892          F = :UAMSEM[UG]
893          UV                                 " V(PMSEM)
894          GOTO(:UFPASWAKING)

895 URROCBANKER(32)
896      U,  $ = UFPAS, P                       " FPAS > 0?
897      Y,  GOTOR(MC[- 1])
898          SUBCD(:UREDDEN)
899          F = - 0                            " MAXSTREAM LENGTH := = 0 AND
900          USTRLEN = F                        " STREAM IDENTIFICATION := NEGATIVE
901          $ = :UTRSTRL1                      " :URLUHANDLELIST
902          MC[0] = $                          " STACK
903          A = MS[0]                          " :URLUHANDLE IN A
904          $ = - 1
905          SUBCD(:UCHANGECLAIM)               " DEC(CLAIM), INC(LOAN), DEC(CASH)
906          $ = MG[4], P                       " EXIST IDLE PLUNCHER CORRECT TYPE?
907      Y,  $ = MA[13], Z                      " NFD = 0?
908      Y,  $ = - MA[14], P                    " NON SELECTED?
909      Y,  $ = MA[11]                         " LUUFD
910      Y,  $ + MA[12]                         " + DESIRE
911      Y,  $ - USTRLEN, P                     " LUUFD + DESIRE  MAX STREAM LENGTH?
912      Y,  SUBCD(:UBANKERSALLOWANCE)          " BANKERSALLOWANCE?
913      Y,  USTRLEN + $                        " MAXSTREAM LENGTH := LUUFD + DESIRE
914      Y,  USTRLEN[1] = A                     " STREAM IDENTIFICATION := :HANDLE
915          $ = 1
916          SUBCD(:UCHANGECLAIM)               " INC(CLAIM); DEC(LOAN); INC(CASH)
917          $ + MC[- 1]
```

Bankers algorithm.

```
918     U,  S - :UTRSTRL1[UNUMPLUSTR], Z    " ALL STREAMS EXAMINED?
919     N,  JUMR(- 18)
920         A = USTRLEN[1], R               " STREAM CHOSEN?
921     Y,  S = - 1
922     Y,  SUBCD(:UCHANGECLAIM)            " DEC(CLAIM), INC(LOAN), DEC(CASH)
923     Y,  SUBCD(:URLUSEL)                 " PLUNCHER SELECTION
924     Y,  SUBCD(:UINCRASLUUED)            " INCREASE FPAS LUUFD
925         SUBCD(:UHERSTEL)
926     Y,  GOTO(:URROCBANKER)              " TRY AGAIN
927         GOTOR(MC[- 1])                  " RETURN

928 UCHANGECLAIM(6)
929         G = MA[16]                      " :UTR(RL)RMVAR
930         MG[0] - S                       " INC OR DEC LOAN
931         MG[1] + S                       " DEC OR INC CLAIM
932         G = MA[15]                      " :UTR(RL)VAR
933         MG[2] + S                       " DEC OR INC CASH
934         GOTOR(MC[- 1])

935 URLUSEL(25)
936         MC[0] = A                       " STACK :HANDLE SELECTED STREAMS
937         MC[0] = - B                     " PLUNCHER IDENTIFICATION := NEGATIVE
938         S = MG[7], R                    " CYCLIC POINTER(:USBTR!)
939         A = 1
940     Y,  MG[4] - A                       " DECREASE NUMBER OF IDLE PLUNCHERS
941         S = UCYCLE[US]
942     Y,  MG[7] = S                       " INCREASE CYCLIC POINTER; S = :USBTR(! + 1)
943         A - UAMVAR[US], Z               " PLUNCHER IDLE?
944     N,  JUMR(6)
945         A = - M[B - 1], R               " PLUNCHER IDENTIFICATION NEGATIVE?
946     Y,  M[B - 1] = S
947         A = - URLUDISC[US], R           " NON DISC?
948     Y,  M[B - 1] = S
949     Y,  A = UCURSTR[US]
950     Y,  A - M[B - 2], Z                 " LAST PLUNCHED FROM CHOSEN STREAM
951     N,  A = UCYCLE[US]
952     N,  A - MG[7], Z                    " ALL PLUNCHERS EXAMINED?
953     N,  JUMR(- 15)
954         G = MC[- 1]                     " :USBTR(RL) CHOSEN
955         UAMVAR[UG] = A                  " PLUVAR := 0
956         A = MC[- 1]                     " :HANDLE
957         UCURSTR[UG] = A                 " CURSTREAM := :HANDLE
958         MA[14] = G                      " SEL := :USBTR(RL)
959         F = :UAMSEM[UG]
960         GOTO(:UVOR[2])                  " V (PLUSEM)

961 URFILLSEGMENT(22)
962         S = MG[10], Z                   " SEGMENT EMPTY?
963         S = :MG[10]                     " ADDRESS PRIVATE SV
964         SUBCD(:UHOLYRMSEG)
965         S = UBLOCKSTR
966         G = MS[10]                      " :CTE[0]
967         G = MG[2]                       " :CRB[0]
968     Y,  A = URUNNUM
969     Y,  MG[0] = A                       " CRB[0] := RUNNUMBER
970     Y,  MG[1] = F                       " NUMBER OF WORDS := 0
971         A = 1
972         RLUSA(MG[1])
973     U,  A - 507, R                      " SEGMENT FILLED?
974         G + A
```

```
 975          A = MS[17]                          " BUILDING UP WORD
 976          MS[17] = A                          " BUILDING UP WORD := - 0
 977          MG[1] = A                           " STORE WORD ON SEGMENT
 978      Y,  A = MT[4]
 979      Y,  MG[2] = A
 980          S = :MS[10]
 981          SUBCD(:UBROEANATION)
 982          GOTO(MC[- 1])                       " RETURN WITH YES CONDITION IF
 983          '776 776 776'                       " SEGMENT FILLED ELSE NO CONDITION

 984 UTREILL(11)
 985          G = UBLOCKSTR
 986          S = MG[17]                          " TAKE BUILDING UP WORD
 987          LCS(9)                              " SHIFT WORD NINE BITS
 988          S '*' - 511
 989          S + A                               " LAST OCTADE IN D[0] THROUGH D[9]
 990          MG[17] = S, P                       " BUILDING UP WORD COMPLETED?
 991      Y,  SUBCD(:UFILLSEGMENT)                " FILL WORD IN SEGMENT
 992      Y,  SUBCD(:USW4)                        " CHAIN ON SEGMENT
 993      Y,  SUBCD(:UTREXVVITADRES)              " WHITE V(UTREX)
 994          G = UBLOCKSTR                       " RESTORE :HANDLE IN G
 995          GOTOR(MC[- 1])                      " RETURN

 996 USW4(23)
 997          UTREXPROOD                          " RED P(UTREX)
 998          A = UBLOCKSTR
 999          S = MA[13], Z                       " NFD = 0?
1000      Y,  S = MA[14], P                       " STREAM SELECTED?
1001      Y,  JUMP(12)
1002          S = UFPAS, Z                        " FPAS = 0?
1003          S = 1
1004      N,  UFPAS - S                           " DEC(FPAS)
1005      N,  MA[11] + S                          " INC(LUUFD)
1006      N,  SUBCD(:CLRSEL)                      " PRINTER SELECTION
1007      Y,  MA[12] = S                          " DESIRE := 1
1008      Y,  S = 11
1009      Y,  UAMVAR = S                          " PMSHR := 11
1010      Y,  SUBCD(:UPROCBANKER)                 " PROCEDURE BANKER
1011      Y,  SUBCD(:UTREXVADRES)                 " V(UTREX)
1012      Y,  SUBCD(:UPOWNSEMADRES)               " P(PMSEM)
1013      Y,  SUBCD(:UTREXPADRES)                 " P(UTREX)
1014          S = 1                               " CURRENT PORTION SIZE
1015          SUBCD(:UFIOBARRIER)                 " FIO BARRIER
1016          S = :MA[10]                         " :SV TO BE CHAINED
1017          SUBCD(:USGAAN)                      " CHAIN ON
1018          G = UBLOCKSTR
1019          GOTO(:UVOP[2])                      " V(STREAM) AND RETURN
1020 USW8(23)
1021          S = MG[3]                           " :SV LAST CHAINED SEGMENT
1022          P = 256, Z
1023          SUBCD(:USGAAN[13])                  " SET FINAL SEGMENT MARKER
1024          A = UBLOCKSTR
1025          S = MA[13], Z                       " NFD = 0?
1026      Y,  S = MA[14], P                       " STREAM SELECTED?
1027      Y,  S = - USRECIALRM, P                 "NORMAL RM?
1028          S = 1
1029          MA[13] + S                          " INC(NFD)
1030          MA[9] + S                           " INC (DOCUMENT NUMBER)
1031          MA[6] = - B                         " DOC. UNDER CONSTRUCTION := FALSE
1032          G = MA[16]                          " :UTR(PL)RMVAR
```

elementary routine
to fill one octad in punches stream.

strategic subroutines,

See EWD 126 & 130

```
1033          MG[2] - S                       " DEC(NUMBER OF DOC. UNDER CONSTR.)
1034     Y,   SUBCD(:UCHANGECLAIM)            " DEC(LOAN), INC(CLAIM), INC(CASH)
1035     Y,   GOTO(:UCASHINCREASE)            " CASH INCREASE
1036          G = MA[15]                      " :UTR(RL)VAR
1037          S = MG[2], Z                    " CASH = 0?
1038     Y,   S = MA[11]                      " LUUFD
1039     Y,   MA[11] - S                      " LUUFD := 0
1040     Y,   MG[0] + S                       " INC(PUNFP OR PLOFP, LUUFD)
1041     Y,   GOTOR(MC[- 1])
1042          SUBCD(:UFINDOCSEL)              " FINISHED DOCUMENT SELECTION
1043          GOTO(:UINCFPASLUUFD)            " INCREASE FPAS WITH LUUFD
1044 UTEXTRLU(28)
1045          UTREXP                          " P(UTREX)
1046          S = USPECIALRLU, P                    "SPECIAL PLUNCHER?
1047     N,   S = 1                                 "PLUNCHER := IDLE
1048          UAMVAR = S
1049     Y,   SUBCD(:UPURM)
1050          G = URLUTYPE
1051     N,   MG[4] + S                             "INC(NUMBER IDLE PLUNCHERS)
1052     U,   MG[2] - S, P                    " CASH > 1?
1053     Y,   SUBCD(:UDISCRLU)                " DISCONNECTION IF WANTED AND ALLOWED
1054          SUBCD(:UFINDOCSEL)              " FINISHED DOCUMENT SELECTION
1055     N,   SUBCD(:UDISCRLU)                " DISCONNECTION IF WANTED AND ALLOWED
1056          SUBCD(:UPROCBANKER)             " UFINISHED DOCUMENT SELECTION
1057          UTREXV                          " V(UTREX)
1058          UPOWNSEM                        " P(PLUSEM)
1059          UTREXP                          " P(UTREX)
1060          S = UAMVAR
1061          S - 3, Z                        " PLUNCHER PREDISCONNECTED?
1062     Y,   SUBCD(:UTREXVADRES)
1063     Y,   SUBCD(:UMES26)                  " MESSAGE M26
1064     Y,   SUBCD(:UTREXVADRES)
1065     Y,   SUBCD(:UPOWNSEMADRES)           " P(PLUSEM)
1066     Y,   GOTO(:UTEXTRLU)                 " IN CIRCULATION AGAIN
1067          S = M3[13], P                   " FINISHED DOCUMENT SELECTED (NFD > 0)?
1068     Y,   S = M3[4]                       " ADDRESS FIRST SV TO BE UNCHAINED
1069     Y,   SUBCD(:USGLONG)                 " COUNT LENGTH OF STREAM
1070          ULENGTHSEG = G
1071     N,   ULENGTHSEG = - 8, E             " NEGATIV INDICATION, LENGTH UNKNOWN
1072          UTREXV                          " V(UTREX)
1073 UTEXTRLU1(38)
1074          G = UCURSTR                     " UPON ENTRY ONLY FIRST TIME YES CONDITION
1075          UP                              " P(STREAMSEM)
1076          UTREXP                          " P(UTREX)
1077          G = UCURSTR                     " :HANDLE
1078          A = :URLUSEG                    " ADDRESS PRIVATE SV TO BE FILLED
1079          SUBCD(:USGAF)                   " UNCHAINING FROM STREAM
1080          ULAST = S                       " BOOLEAN LAST SEGMENT
1081          S = 1
1082          UFIO + S                        " INC(FIO)
1083          UFIOP + S                       " INC(FIOP)
1084          SUBCD(:UFIOFIOP)                " USE FIO AND FIOP FOR WAKING UP
1085          UTREXV                          " V(UTREX)
1086     Y,   S = :URLUSEG                    " ADDRESS PRIVATE SV
1087     Y,   SUBCD(:SRQ)                     " ASK FIRST SEGMENT HOLY IN CORE
1088     Y,   A = M5[2]                  " :CRB[0] PRIVATE SEGMENT
1089     Y,   A = MA[0]                       " RUNNUMBER ON SEGMENT
1090     Y,   UEXNUMDOC = A                   " STORE IN PLUNCHERBOTTOM
1091     Y,   SUBCD(:URPOFANATION)            " PROFANATION PRIVATE SEGMENT
1092     Y,   DO(M6[8])                       " BEGIN PLUNCHERDOCUMENT
```

```
1093          DO(M6[9])                          " PROCES SEGMENT
1094          G = M5[2]                                ":CRB[0] PRIVATE SEGMENT
1095          S = :URLUSEG
1096          $UBCD(:UKILLSEG)                   " KILL SEGMENT
1097          S = ULAST, R                       " WAS THIS THE LAST SEGMENT?
1098      N,  GOTO(:UTEXTRLU1)                   " NEXT SEGMENT, NO CONDITION
1099          DO(M6[10])                         " END OF DOCUMENT
1100          S = - USRECIALRLU, R                     "NORMAL PLUNCHER?
1101      Y,  S = - M1[0]
1102      Y,  S '*' - UCON2, Z                         "NBK?
1103      Y,  $UBCD(:UMES28)                     " MESSAGE M28 (DOCUMENT NOT CORRECT PROCESSED)
1104          UTREXR                             " R(UTREX)
1105      Y,  S = - URLUDISC, R                  " DISC = FALSE?
1106          S = 1
1107      Y,  URLUDISC = B                       " DISC := TRUE
1108      Y,  M6[1] - S                          " PLUACCEPT := PLUACCEPT - 1
1109          M3[13] - S                         " DEC(NFD)
1110          M3[14] = - B                       " STREAMSELECTED := EALSE
1111          GOTO(:UTEXTRLU[1])                 " LOOK FOR NEW JOB
1112 UPUPM(13)
1113          $UBCD(:UEXPMTRVAR)
1114          MC = G
1115      Y,  F = :ULISTTR
1116      Y,  SUBCD(:UEXPMTRVAR[1])
1117          S = MC[- 1]
1118      N,  GOTOR(MC[- 1])
1119          A = - 0
1120          UAMVAR[UG] = A
1121          UAMVAR[US] = A
1122          A = UBLOCKSTR[US]
1123          UCURSTR[UG] = A
1124          MA[14] = G
1125          GOTO(:UHAR1[3])
1126 ULISTTR(UNUMTR[2])
1127          :ULISTTR[2]
1128          UNUMTR
1129          :USBTRA
1130          :USBTRB
1131          :USBTRC

1132 UBEGINTRDOC(25)        begin of tape punch document
1133          S = USRECIALRLU,R
1134      Y,  JUMP(16)
1135          S = ULENGTHSEG, R                  " IF POSITIV LENGTH IN SEGMENTS
1136      Y,  MULS(- USTMAXHERSEG)               " TIMES - MAX. NUMBER HEPTADES PER SEGMENT
1137      N,  S = - USTHERPLENTY                 " - PLENTY HEPTADES(SAY 1/4 REEL)
1138          S + UAVAILTAPE, R                  " SUFFICIENT AVAILABLE TAPE?
1139      N,  SUBCD(:UMES20)                     " MESSAGE M20(ASK FOR NEW REEL)
1140      N,  JUMP(5)
1141          S = UCUTVAR                        " :HANDLE STREAM FROM WHICH LAST PUNCHED
1142          S - UCURSTR, Z                     " IS THIS CURRENT STREAM?
1143      Y,  S = UEXNUM                         " RUNNUMBER PROGRAM FOR WHICH LAST PUNCHED
1144      Y,  S - UEXNUMDOC, Z                   " DOCUMENT FROM SAME PROGRAM?
1145      N,  SUBCD(:UMES23)                     " MESSAGE M23(TEAR OFF TAPE)
1146      N,  SUBCD(:UPOWNSEMADRES)              " R(PUNSEM)
1147      N,  S = UCURSTR
1148      N,  UCUTVAR = S                        " CUTVAR := CURSTREAM
1149      N,  S = UEXNUMDOC
1150      N,  UEXNUM = S                         " UPDATE RUNNUMBER
1151          SUBCD(:UHOKPLUNCHER[3])                  " NOK COMMAND, V(AET), R(IET)
```

```
1152          $ = UMOKBLU, R                    " OK?
1153      N,  JUMP(- 3)
1154          $ = UNUMBLANK1
1155          SUBCD(:UBLANKTAPE)                " PUNCH BLANK TAPE
1156          G = UTRLETTER                     " INVARIANT ADDRESS PUNCH IDENTIFICATION
1157          GOTO(:URUNCHSTRING)

1158 UTRSEGMENT(4)
1159          F = :URLUSEG
1160          F * 512
1161          F + 2                             " G IS INV. ADDRESS FIRST HEPTADE ON SEGMENT
1162          GOTO(:URUNCHSTRING)

1163 UENDTRDOC(6)
1164          $ = UNUMBLANK2
1165          SUBCD(:UBLANKTAPE)
1166          G = UALLHOLE
1167          SUBCD(:URUNCHSTRING)
1168          $ = UNUMBLANK3
1169          GOTO(:UBLANKTAPE)

1170 UBLANKTAPE(7)
1171          MC[0] = S
1172          G = UBLANK
1173          SUBCD(:URUNCHSTRING)
1174          $ = MC[- 1]
1175          $ - 1, Z
1176      N,  JUMP(- 6)
1177          GOTOR(MC[- 1])

1178 URUNCHSTRING(68)
1179          SUBCD(:UINSTV)                    " INITIALIZE STRING VARIABLE
1180          MC[0] = E                         " SET SOME BOOLEAN TRUE
1181          E = :MC[- 5]                      " :STV[0]
1182          A = M4[1]
1183          A '*' UCON2
1184          M4[1] = A                         " INITIALIZE CODEWORD
1185          SUBCD(:UDNSS)                     " NEXT HEPTADE IN A
1186      U,  A - 510, Z                        " END OF STRING?
1187      Y,  JUMP(10)
1188          M[B - 1] = - B                    " ALL HEPTADES PUNCHED := EALSE
1189          S = UCON1
1190          PLUSS(M4[1])                      " INCREASE COUNT PART CODEWORD
1191          RUS(18)
1192      U,  $ - UTPLEN, Z                     " BUFFER FILLED?
1193          S + M4[1]
1194          MS[0] = A                         " STORE HEPTADE IN BUFFER
1195          S = 1
1196          UAVAILTAPE - S                    " DECREASE AVAILABLE TAPE
1197      N,  JUMP(- 13)
1198          $ = M[B - 1], R                   " ALL HEPTADES PUNCHED?
1199      Y,  B - 5
1200      Y,  GOTOR(MC[- 1])
1201          $ = M4[1]
1202          S '*' - UCON2, Z                  " LAST FILLED BUFFER EMPTY
1203          S = ULALAB
1204      Y,  JUMP(4)
1205          SUBCD(:UTEXTART)                  " V(ART), NEW ULALAB
1206          A = M[B - 2], R                   " FIRST STARTING COMMAND?
1207      Y,  M[B - 2] = - B
```

```
1208      Y,  JUMP(- 28)                    " FILL OTHER BUFFER
1209          G = MD[2]
1210          URH                           " R(IFT)
1211      Y,  M[B - 1] = B                  " ALL HEPTADES PUNCHED := IRUE
1212      Y,  S = MS[0]
1213      Y,  ULALAB = S                    " ULALAB POINTS TO LAST FINISHED COMMAND
1214          A = M4[- 1], R                " CLOSING WORD
1215          LCA(8), E                     " NOT E,B?
1216      N,  A = URUNARROW, R              " PUNCHING ARROW?
1217      Y,  JUMP(- 37)
1218          A = M[B - 1], R               " ALL HEPTADES PUNCHED?
1219      N,  SUBCD(:UROPH)
1220          SUBCD(:UHOKRLUNCHER)          " HOK, V(AFT), R(IFT)
1221          SUBCD(:UTEXTAFT)              " PUNCH IN ASL SITUATION
1222      N,  SUBCD(:UTEXTAFT)
1223          URH
1224      N,  SUBCD(:UROPH)
1225          M[B - 2] = E                  " SET BOOLEANS TRUE
1226          URUNARROW = B                 " PUNCHING ARROW := IRUE
1227          GOTO(:UNEWARROW1)
1228          SUBCD(:URUNCHSTRING)
1229          S = UNUMBLANK4
1230          SUBCD(:UBLANKTAPE)
1231          SUBCD(:UHOKRLUNCHER)          " SITUATION OK RED
1232          S = USPECIALRLU, R
1233      N,  SUBCD(:UMES20)
1234      N,  SUBCD(:UROWNSEMADRES)
1235          G = UARROW2
1236          SUBCD(:URUNCHSTRING)
1237          S = UNUMBLANK5
1238          SUBCD(:UBLANKTAPE)
1239          G = UALLHOLE4
1240          SUBCD(:URUNCHSTRING)
1241          A = M1[0], R                  " AR0
1242          LCA(8), E                     " NOT E,B?
1243      N,  SUBCD(:UHOKRLUNCHER)
1244      N,  JUMP(- 14)
1245          URUNARROW = - B
1246          GOTO(:URUNCHSTRING[2])        " CONTINE WITH ORIGINAL STRING
1247 UNEWARROW1(6)
1248          G = UALLHOLE4
1249          SUBCD(:URUNCHSTRING)
1250          S = UNUMBLANK5
1251          SUBCD(:UBLANKTAPE)
1252          G = UARROW2
1253          GOTO(:URUNCHSTRING[49])
1254 UHOKRLUNCHER(5)
1255          A = 10000
1256          A - 1, Z
1257      N,  JUMP( - 2)
1258          A = :UHOKRLU[1]
1259          GOTO(:CVAETRIET[1])
1260 UMES20(7)
1261          SUBCD(:UCLAIMCON)             " CLAIM CONSOLE
1262          UTREXR
1263          S = 2                         " PUNVAR := 2
1264          UAMVAR = S
1265          UTREXV
1266          G = UM20                      " UINVM20A/B
1267          GOTO(:UMES8[4])               " MERGE WITH MESSAGE M8
```

```
1268 UAN$21(13)
1269          UTREXR
1270          G = UMIREAC[5]                  " :USBTRA/B
1271          S = UAMVAR[UG]
1272          $ - 2, Z                        " PUNVAR = 2?
1273      Y,  UAMVAR[UG] = S                  " PUNVAR := 0
1274      Y,  $ = USTHERREEL                  " NUMBER OF HEPTADES PER REEL
1275      Y,  UAVAILTAPE[UG] = $              " AVAILABLE TAPE := ONE REEL
1276      Y,  R = :UAMSEM[UG]
1277      Y,  SUBCD(:UVOR[2])                 " V(PUNSEM)
1278      Y,  GOTO(:UENDMES[3])
1279          UTREXV
1280          G = CMNA21                      " NOT APPLICABLE
1281          GOTO(:UENDMES)
1282 UME$23(5)
1283          SUBCD(:UCLAIMCON)               " CLAIM CONSOLE
1284          G = UM23
1285          SUBC(:CPRINTSTRING)             " PRINT MESSAGE M23
1286          S = :CPLUMES[16]                " :PROPER CODE LIST
1287          GOTO(:UCONCON)                  " CONDITION CONSOLE FOR ANSWER M24 I.E.
1288                                          " UMIREAC[0] = :PROPER CODE LIST
1289                                          " UMIREAC[1] = :USBTRA/B

1290 UME$25(13)
1291          UTREXR
1292          G = UMIREAC[5]
1293          A = UAMVAR[UG]
1294      U,  A - 4, Z                        " PUNVAR = 4?
1295      Y,  GOTO(:UME$9[9])                 " ALREADY DISCONNECTED
1296          A - 3, Z                        " PUNVAR = 3?
1297      N,  A = URLUDISC[UG], R             " DISCONNECT TRUE?
1298      N,  URLUDISC[UG] = B                " DISCONNECT := TRUE
1299      N,  A = 1
1300      N,  S = URLUTYRE[UG]                " :UTR(RL)VAR
1301      N,  MS[1] - A                       " DEC(PUN OR PLO ACCEPT)
1302      N,  SUBCD(:UDISCRLU)                " DISCONNECT PLUNCHER
1303          GOTO(:UENDMES[3])

1304 UME$26(5)
1305          SUBCD(:UCLAIMCON)               " CLAIM CONSOLE
1306          UTREXR
1307          $ = 4
1308          UAMVAR = S                      " PLUVAR := 4
1309          GOTO(:UME$12[4])                " MERGE WITH MESSAGE M12

1310 UME$27(13)
1311          UTREXR
1312          S = UMIREAC[5]                  " :USBTR(RL)A/B
1313          A = UAMVAR[US]
1314          A - 4, Z                        " PLUVAR = 4?
1315      N,  GOTO(:UME$13[7])                " N.A. MERGE WITH MESSAGE M13
1316          UAMVAR[US] = A                  " PLUVAR := 0
1317          G = URLUTYRE[US]                " :UTR(RL)VAR
1318          A = 1
1319          MG[1] + A                       " PLUACCEPT := PLUACCEPT + 1
1320          MG[2] + A                       " PLUCASH := PLUCASH + 1
1321          SUBCD(:UCASHINCREASE)           " CASH INCREASE
1322          S = UMIREAC[5]                  " :USBTR(RL)A/B
1323          GOTO(:UAN$2[7])                 " MERGE WITH UAN$2
```

```
1324 UMES28(6)
1325          SUBCD(:UCLAIMCON)
1326          G = UM28
1327          SUBC(:CPRINTSTRING)
1328          A = UEXNUMDOC
1329          SUBC(:CPRINTDEC)
1330          GOTO(:UCONFREE)

1331 UBEGINPLDOC(12)    begin of plotter document
1332          SUBCD(:UMES20)
1333          UPOWNSEM ------> P(ownsem)
1334          SUBCD(:UHOKPLUNCHER)
1335          UFIRSTPL = B
1336          S = UPLOWIDTH[1]
1337          A = :UPENUR[1]
1338          SUBCD(:CVAETRIET[1])
1339          A = :UPENLINKS[1]
1340      U,  S = - M1[0], R                    " NOT OK?
1341      N,  S - 1, Z
1342      N,  JUMP(- 5)
1343          GOTO(:UHOKPLUNCHER)
1344 UPLCODE(8)
1345          1
1346          2
1347          4
1348          8
1349          5
1350          9
1351          6
1352          10
1353 UPENUR(4)
1354          -0
1355          -0
1356          :UPENUR[UCONN1 + 2]
1357          16
1358 UPENLINKS(4)
1359          -0
1360          -0
1361          :UPENLINKS[UCONN1 + 2]
1362          4
1363 USCHAKELPLA(6)
1364          -0
1365          :USCHAKELPLA[4]
1366          :UWIGPLA[UPLLENTEL - 1]
1367          -0
1368          :USCHAKELPLA[1]
1369          :UWAGPLA[UPLLENTEL - 1]
1370 UANSPL21(10)
1371          UTREXR
1372          G = UMIREAC[5]
1373          S = UAMVAR[UG]
1374          S - 2, Z

1375      Y,  UAMVAR[UG] = S
1376      Y,  GOTO(:UANS21[7])
1377          UTREXV
1378          G = MT[1]
1379          GOTO(:UENDMES)
1380          CINVRLM21
```

platter routines
permanent core occupancy

```
1381 UIFTRLA(10)
1382         :UARRLA
1383         -0
1384         -0
1385         '760 071 000'[4]
1386         '660 071 000'[4]
1387         '760 072 000'[4]
1388         '660 072 000'[4]
1389         '760 070 000'[4]
1390         '000 020 000'
1391         0

1392 UEINDRITEN(2)                          " THROWN OUT BY OPERATOR
1393         A = - 0
1394         GOTO(:UDYNERR[8])              " MERGE WITH DYNAMIC ERROR
1395 UDYNERR(21)                           " ENTRY POINT DYNAMIC ERROR
1396         S = MC[- 1]
1397         UDEAE
1398         GL3 = A                        " SAVE FAILLURE NUMBER
1399         GL4 = S                        " SAVE LINK
1400         A = 1
1401     U,  A - UKILLVAR, Z                " KILLVAR = 1?
1402     Y,  S = UKILLCOUNTER, Z            " KILLCOUNTER ZERO?
1403     N,  GOTO(:USTERR)                  " SYSTEM ERROR
1404         UKILLCOUNTER = A               " ENTRY POINT OPERATOR EXIT
1405         S = UMAMR[1], Z                " WHITE?
1406     N,  A = 2
1407     N,  GOTO(:USTERR)                  " SYSTEM ERROR
1408         B = :GL20
1409         F = :UEXSEM                    " EXCLUSION SEMAPHORE IN PROGRAM EXIT
1410         UR                             " R(EXSEM)
1411         B = :UWSEXIT                   " UNIVERSAL WORKING SPACE IN PROGRAM EXIT
1412         S = GL26
1413         GL6 = S                        " FOR FUTURE PROFANATION
1414         G = MT[1]
1415         GOTO(:R46E3)
1416         UINVEXIT
1417 ONZIN9(2)
1418         A = 205
1419         GOTO(:UDYNERR)                 "INDEX OUT OF BOUNDS, ONE DIMENSIONAL ARRAY
1420 ONZIN11(3)
1421         B + 1
1422         A = 204                        "ENTRY ONZIN5, 8, 10
1423         GOTO(:UDYNERR)                 "DIMENSION ERROR
1424 ONZIN12(5)
1425         S = UWS[0]
1426         M[B] = S
1427         B + 1                          "ENTRY ONZIN6, 7
1428         A = 206
1429         GOTO(:UDYNERR)                 "INDEX OUT OF BOUNDS MORE DIMENSIONAL ARRAY
1430 RSE56(2)
1431         A = 227
1432         GOTO(:UDYNERR)
1433 RSE64(2)
1434         A = 226
1435         GOTO(:UDYNERR)
1436 RSE116(2)
1437         A = 202
1438         GOTO(:UDYNERR)                 "ARRAY BY VALUE
```

dynamic errors

```
ONZ!N1(2)
        S = :UDISASTER1[1]
        CRLABEL[1] = S
ONZ!N3(0)
ONZ!N4(10)
        S = CRLABEL[1]
        A = UCON1
        RLUSA(UARCR[2])
        RUA(19), Z
    Y,  A = UARCR
    Y,  A '*' UCON2
    Y,  A = S
    Y,  UARCR = A
    Y,  DO(UIETCR[7])
        JUMP(- 1)
UDISASTER1(21)
        - 0
        - 0
        :UDISASTER1['022 000 002']
        36                          " LETTERS, BLACK
        2                           " CARRIAGE RETURN
        8                           " NEW LINE
        18                          " D
        10                          " R
        28                          " U
        7                           " M
        4                           " SPACE
        18                          " D
        12                          " I
        20                          " S
        3                           " C
        10                          " R
        18                          " D
        16                          " E
        10                          " R
        32                          " DIGITS, BLACK
        7                           " DOT

UDISASTER2(24)
        - 0
        - 0
        :UDISASTER2['025 000 002']
        36                          " LETTERS, BLACK
        2                           " CARRIAGE RETURN
        8                           " NEW LINE
        7                           " M
        16                          " E
        7                           " M
        3                           " C
        10                          " R
        21                          " Y
        4                           " SPACE
        16                          " E
        23                          " X
        5                           " H
        24                          " A
        28                          " U
        20                          " S
        1                           " T
        16                          " E
```

" DISASTER SEGMENT CONTROLLER

" DISASTER USW1
" DISASTER UTEXTTR

" 2 ↑ 18
" AFT := AFT + 1
" AFT = 1?
" OLD ARO
" ISOLATE ADDRESS PART
" OLD ADDRESS PART - NEW ADDRESS PART
" NEW ADDRESS PART IN ARO
" AF := TRUE

```
1498          18                                  " 0
1499          32                                  " DIGITS, BLACK
1500          7                                   " DOT

1501 UD200(5)
1502          SUBCD(:SRQ)
1503          G = MS[0]
1504          $ = MG[2]
1505          $ + MC[-1]
1506          GOTO(:UDYNERR[1])
1507 USTERR(2)
1508          UDEAF
1509          JUMP(- 1)
1510 UEXSEM(3)
1511          + 1
1512          - 0
1513          - 0
1514 URUNSEM(3)
1515          + 1
1516          - 0
1517          - 0
1518 URUNAAN(1)
1519          UINVRUN1[- 1]

1520 URACKEDCODE(10)
1521          0                        " + X
1522          1                        " - X
1523          43                       " PEN UP
1524          2                        " + Y
1525          4                        " + X AND + Y
1526          6                        " - X AND + Y
1527          44                       " PEN DOWN
1528          3                        " - Y
1529          5                        " + X AND - Y
1530          7                        " - X AND - Y
1531 UMOVE(55)
1532          G = UBLOCKSTR
1533          A + :URACKEDCODE[- 1]
1534          $ = MA
1535          A = 1, Z
1536          MG[22] + A
1537          JUMP($)
1538          JUMP(24)                 " + X
1539          JUMP(6)                  " - X
1540          JUMP(9)                  " + Y
1541          JUMP(17)                 " - Y
1542          JUMP(11)                 " + X AND + Y
1543          JUMP(14)                 " + X AND - Y
1544          JUMP(4)                  " - X AND + Y
1545          MG[20] + A, R            " - X AND - Y
1546     N,   MG[19] + A, R
1547     N,   JUMP(16)
1548          JUMP(12)
1549          MG[19] + A, R
1550          MINA(MG[20])
1551     N,   A + URLOWIDTH[1], Z
1552     N,   JUMP(11)
1553          JUMP(7)
1554          MG[19] - A
1555          MINA(MG[20])
```

Subroutine elementary plotter step

```
1556          A + URLOVIDTH[1], Z
1557      N,  JUMP(6)
1558      N,  MG[19] - A

1559      N,  MG[20] + A, P
1560      N,  JUMP(3)
1561          A = 236
1562          GOTO(:UDYNERR)
1563          MG[19] - A
1564          MC = S
1565          S = MG[17]
1566          A = MG[18]
1567          LCSA(3), P
1568          A + MC[- 1]
1569          MG[18] = A
1570          MG[17] = S
1571      Y,  DO(MD[-1])
1572      Y,  GOTOR(MC[- 1])
1573          A = - MG[19]              " X - COORDINATE
1574      U,  A - MG[21], P             " X > XMAX?
1575      Y,  MG[21] = A                " NEW XMAX
1576          SUBCD(:UFILLRLSEG)
1577      Y,  SUBCD(:USH4)
1578      Y,  SUBCD(:UTREXVWITADRES)
1579          DO(MD[- 1])
1580          GOTO(:UMORTAL)
1581          S = MT[3], P              " ENTRY PEN UP
1582      N,  S = MT[3]                 " ENTRY PEN DOWN
1583          MG[17] + S
1584          JUMP(- 9)
1585          '100 000 000'
1586          '200 000 000'

1587 UFILLRLSEG(26)
1588          SUBCD(:UIMMORTAL)
1589          S = MG[10], Z             " SEGMENT EMPTY?
1590          S = :MG[10]
1591          SUBCD(:UHOLYRMSEG)
1592          S = UBLOCKSTR
1593          G = MS[10]
1594          G = MG[2]
1595      Y,  A = URUNNUM
1596      Y,  MG[0] = A
1597      Y,  MG[1] = E
1598          A = MS[21]                " MAXIMUM X - COORDINATE
1599          A + MS[19]                " -CURRENT X - COORDINATE
1600          MG[2] = A
1601          A = 2
1602          PLUSA(MG'1])              " INCREASE NUMBER OF WORDS ON SEGMENT
1603      U,  A - 506, Z                " SEGMENT FILLED?
1604          A + G
1605          E = MS[17]
1606          MA[1] = E                 " FILL BUILDING UP WORD
1607          E = 4
1608          MS[17] = E                " INITIALIZE BUILDING UP WORD
1609          E = 0
1610          MA[3] = E                 " ENDMARKER
1611          S = :MS[10]
1612          SUBCD(:UPROFANATION)
1613          GOTO(MC[- 1])
```

```
1614 URLSEGMENT(35)
1615          A = :URLUSEG
1616          LCA(9)
1617          A + 3
1618          UINVRLADRES = A, Z
1619          SUBCD(:RSE41)
1620          S = E, Z
1621     Y,   GOTOR(MC[- 1])
1622          URACKEDRLWORD = S, R
1623          A = M[58]
1624          E = 17
1625     Y,   E - 1
1626     Y,   LCSA(3), R
1627     Y,   JUMR(- 3)
1628          LCSA(3)
1629          E + 0, Z
1630     N,   LCSA(3)
1631     N,   MC = S
1632     N,   MC = A
1633     N,   A '*' 7
1634     N,   A + :URLCODE
1635     N,   A = MA[0]
1636     N,   SUBCD(:URLSTAR)
1637     N,   A = MC[- 1]
1638     N,   S = MC[- 1]
1639     N,   E - 1, Z
1640     N,   JUMR(- 11)
1641          S = URACKEDRLWORD
1642          LCS(1), R
1643     N,   A = 32
1644     Y,   A = 16
1645     Y,   LCS(1), R
1646     N,   SUBCD(:URLSTAR)
1647          A = 2
1648          PLUSA(UINVRLADRES), Z
1649          JUMR(- 31)

1650 URLSTAR(15)
1651          S = 1
1652          PLUSS(URLIN), Z
1653          S + M4[1]
1654          MS[URLLEN] = A
1655     N,   GOTOR(MC[- 1])
1656          SUBCD(:UREDDEN)
1657          S = - URLLEN
1658          URLIN = S
1659          S = MD[4]
1660          SUBCD(:UTEXTAFT)
1661          S = UEIRSTRL, R
1662     Y,   UEIRSTRL = - B
1663     N,   E = :M2[0]
1664     N,   SUBCD(:UBORH)
1665          GOTO(:UHERSTELTERUG)

1666 UENDRLDOC(14)
1667          G = MG[2]
1668          E + URLRUNOUT
1669          A = 1
1670          SUBCD(:URLSTAR)
```

```
1671          R - 1, Z
1672      N,  JUMR(- 4)
1673          E = :M2[0]
1674          URH
1675          S = M1[0], R
1676          LCS(8), E
1677      Y,  GOTOR(MC[- 1])
1678          SUBCD(:UCLA!MCON)
1679          G = UM23
1680          GOTO(:UMES28[2])

1681 U$VLIB(100)
1682          - 0
1683          - 0
1684          - 0
1685          - 0
1686          - 0
1687          - 0
1688          - 0
1689          - 0
1690          - 0
1691          - 0
1692          - 0
1693          - 0
1694          - 0
1695          - 0
1696          - 0
1697          - 0
1698          - 0
1699          - 0
1700          - 0
1701          - 0
1702          - 0
1703          - 0
1704          - 0
1705          - 0
1706          - 0
1707          - 0
1708          - 0
1709          - 0
1710          - 0
1711          - 0
1712          - 0
1713          - 0
1714          - 0
1715          - 0
1716          - 0
1717          - 0
1718          - 0
1719          - 0
1720          - 0
1721          - 0
1722          - 0
1723          - 0
1724          - 0
1725          - 0
1726          - 0
1727          - 0
1728          - 0
1729          - 0
```

100 SV's for library segments
Initially these SV's indicate empty segments.
On reading in the system these SV's will gradually take their new value.
as library texts are read in and their segments filled.

```
1730          - 0
1731          - 0
1732          - 0
1733          - 0
1734          - 0
1735          - 0
1736          - 0
1737          - 0
1738          - 0
1739          - 0
1740          - 0
1741          - 0
1742          - 0
1743          - 0
1744          - 0
1745          - 0
1746          - 0
1747          - 0
1748          - 0
1749          - 0
1750          - 0
1751          - 0
1752          - 0
1753          - 0
1754          - 0
1755          - 0
1756          - 0
1757          - 0
1758          - 0
1759          - 0
1760          - 0
1761          - 0
1762          - 0
1763          - 0
1764          - 0
1765          - 0
1766          - 0
1767          - 0
1768          - 0
1769          - 0
1770          - 0
1771          - 0
1772          - 0
1773          - 0
1774          - 0
1775          - 0
1776          - 0
1777          - 0
1778          - 0
1779          - 0
1780          - 0
1781          - 0

1782 USBRLA(24)
1783          :UIETRLA
1784          - 0                              " UCURSTR
1785          :USCHAKELRLA[1]                  " ULALAB
1786          - 0                              " URLUSEG
1787          :URLVAR                          " URLUTYRE
1788          - 0                              " URLUDISC
```

```
1789          UINVRLM26A                    " UM12
1790          - 0                           " UWOKRLU
1791          - 0
1792          + 0
1793          :USBRLA                       " UCYCLE
1794          UINVRLM28A                    " UM28
1795          - 0                           " ULAST
1796          - 0                           " UEXNUM
1797          - 0                           " UEXNUMDOC
1798          - 0                           " ULENGTHSEG
1799          - 0                           " UFIRSTRL
1800          - 0                           " UINVRLADRES
1801          - 0                           " UPACKEDRLWORD
1802          - 0
1803          UINVRLM20A                    " UM20
1804          UINVRLM23A                    " UM23
1805          - 0                           " USPECIALRLU
1806          - URLLEN                      " URLIN

1807 URLHANDLE11(23)
1808          - 0                                 " STREAMSEMAPHORE
1809          - 0
1810          - 0
1811          - 0                                 " AANHAAKLABEL
1812          - 0                                 " AFHAAKLABEL
1813          :USBRM1['1 000 000']                " BOTTOM RM
1814          - 0                                 " DOCUMENT UNDER CONSTRUCTION
1815          - 0                                 " ATTACHMENT TIME
1816          - 0
1817          + 1                                 " DOCUMENT NUMBER IN CURRENT PROGRAM OF RM
1818          - 0                                 " SV FOR RM SEGMENT
1819          - 0                                 " LUUFD
1820          - 0                                 " DESIRE
1821          - 0                                 " NFD
1822          - 0                                 " STREAM NOT SELECTED
1823          :URLVAR
1824          :USBRM1[29]
1825          + 0                                 " HEAD BUILDING UP WORD
1826          + 4                                 " TAIL BUILDING UP WORD
1827          - 0                                 " X COORDINATE NEGATIV
1828          - URLOWIDTH                         " Y COORDINATE NEGATIV
1829          - 0                                 " MAXIMAL X COORDINATE
1830          - 0                                 " TOTAL NUMBER OF PLOTTER STEPS

1831 URLHANDLE21(23)
1832          - 0                                 " STREAMSEMAPHORE
1833          - 0
1834          - 0
1835          - 0                                 " AANHAAKLABEL
1836          - 0                                 " AFHAAKLABEL
1837          :USBRM2['1 000 000']                " BOTTOM RM
1838          - 0                                 " DOCUMENT UNDER CONSTRUCTION
1839          - 0                                 " ATTACHMENT TIME
1840          - 0
1841          + 1                                 " DOCUMENT NUMBER IN CURRENT PROGRAM OF RM
1842          - 0                                 " SV FOR RM SEGMENT
1843          - 0                                 " LUUFD
1844          - 0                                 " DESIRE
1845          - 0                                 " NFD
1846          - 0                                 " STREAM NOT SELECTED
```

more stackbottoms and handles

```
1847          :URLVAR
1848          :USERM2[29]
1849          + 0                                   " HEAD BUILDING UP WORD
1850          + 4                                   " TAIL BUILDING UP WORD
1851          - 0                                   " X COORDINATE NEGATIV
1852          - URLOWIDTH                           " Y COORDINATE NEGATIV
1853          - 0                                   " MAXIMAL X COORDINATE
1854          - 0                                   " TOTAL NUMBER OF PLOTTER STEPS

1855 URLHANDLE31(23)
1856          - 0                                   " STREAMSEMAPHORE
1857          - 0
1858          - 0
1859          - 0                                   " AANHAAKLABEL
1860          - 0                                   " AFHAAKLABEL
1861          :USERM3['1 000 000']                  " BOTTOM PM
1862          - 0                                   " DOCUMENT UNDER CONSTRUCTION
1863          - 0                                   " ATTACHMENT TIME
1864          - 0
1865          + 1                                   " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
1866          - 0                                   " SV FOR PM SEGMENT
1867          - 0                                   " LUUFD
1868          - 0                                   " DESIRE
1869          - 0                                   " NFD
1870          - 0                                   " STREAM NOT SELECTED
1871          :URLVAR
1872          :USERM3[29]
1873          + 0                                   " HEAD BUILDING UP WORD
1874          + 4                                   " TAIL BUILDING UP WORD
1875          - 0                                   " X COORDINATE NEGATIV
1876          - URLOWIDTH                           " Y COORDINATE NEGATIV
1877          - 0                                   " MAXIMAL X COORDINATE
1878          - 0                                   " TOTAL NUMBER OF PLOTTER STEPS

1879 URLHANDLE41(23)
1880          - 0                                   " STREAMSEMAPHORE
1881          - 0
1882          - 0
1883          - 0                                   " AANHAAKLABEL
1884          - 0                                   " AFHAAKLABEL
1885          :USERM4['1 000 000']                  " BOTTOM PM
1886          - 0                                   " DOCUMENT UNDER CONSTRUCTION
1887          - 0                                   " ATTACHMENT TIME
1888          - 0
1889          + 1                                   " DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
1890          - 0                                   " SV FOR PM SEGMENT
1891          - 0                                   " LUUFD
1892          - 0                                   " DESIRE
1893          - 0                                   " NFD
1894          - 0                                   " STREAM NOT SELECTED
1895          :URLVAR
1896          :USERM4[29]
1897          + 0                                   " HEAD BUILDING UP WORD
1898          + 4                                   " TAIL BUILDING UP WORD
1899          - 0                                   " X COORDINATE NEGATIV
1900          - URLOWIDTH                           " Y COORDINATE NEGATIV
1901          - 0                                   " MAXIMAL X COORDINATE
1902          - 0                                   " NUMBER OF PLOTTER STEPS
1903 URLHANDLE51(23)
1904          -0                                    "STREAMSEMAPHORE
```

```
1905          -0
1906          -0
1907          -0                                  "AANHAAKLABEL
1908          -0                                  "AFHAAKLABEL
1909          :USBPM5['1 000 000']                "BOTTOM PM
1910          -0                                  "DOCUMENT UNDER CONSTRUCTION
1911          -0                                  "ATTACKMENT TIME
1912          -0
1913          +1                                  "DOCUMENT NUMBER IN CURRENT PROGRAM OF PM
1914          -0                                  "SV FOR PM SEGMENT
1915          -0                                  "LUUFD
1916          -0                                  "DESIRE
1917          -0                                  "NFD
1918          -0                                  "STREAM NOT SELECTED
1919          :UPLVAR
1920          :USBPM5[29]
1921          + 0                                 " HEAD BUILDING UP WORD
1922          + 4                                 " TAIL BUILDING UP WORD
1923          - 0                                 " X COORDINATE NEGATIV
1924          - UPLOWIDTH                         " Y COORDINATE NEGATIV
1925          - 0                                 " MAXIMAL X COORDINATE
1926          - 0                                 " NUMBER OF PLOTTER STEPS

1927 UBIT(15)
1928              A=MC[-2]
1929              M[B-1]=G
1930              G=B
1931              B=M[B-1],R
1932         N,   B=-B
1933              S=0
1934         U,   B-26,R
1935         Y,   B-27
1936         Y,   JUMR(-3)
1937              RCA(B+1)
1938              LCSA(1)
1939              B=G
1940              G=S
1941              DO(MD[-1])
1942              GOTOR(MC)

1943 USHIFT(26)
1944              S=G
1945              A = MC[- 2]
1946              F = :MC
1947              B = MC[- 2]
1948         U,   B-0,R
1949         Y,   JUMR(9)
1950              B=-B
1951         U,   B-26,R
1952         Y,   B-27
1953         Y,   JUMR(-3)
1954         U,   A-0,Z
1955         Y,   RCS(B)
1956         N,   A=0
1957         N,   RCSA(B)
1958              JUMR(7)
1959         U,   B-26,R
1960         Y,   B-27
1961         Y,   JUMR(-3)
1962         U,   A-0,Z
```

logical subroutines

this is one of the few cases where B doesn't serve as stack pointer, but shifting is done under control of B. (Poor decision in the hardware design)

They have nothing to do with boolean expressions, but cater for the user being able to write things like AND(x,1) to test for odd

BIT(n,7) to test for certain bit in an integer

```
1963      Y,  LCS(B)
1964      N,  A=0
1965      N,  LCSA(B)
1966          B=:MG[-1]
1967          G=S
1968          DO(MD[-1])
1969          GOTOR(MC[1])

1970 UOR(6)
1971          S=MC[-2]
1972          A=-G
1973          A'*'-S
1974          G=-A
1975          DO(MD[-1])
1976          GOTOR(MC)

1977 UAND(6)
1978          S=MC[-2]
1979          A=G
1980          A'*'S
1981          G=A
1982          DO(MD[-1])
1983          GOTOR(MC)

1984 UCOMPOSE(10)
1985          A=M[B-2]
1986          S=G
1987      U,  JUMP(0)
1988          E=A
1989          E+0
1990          A = 248
1991      U,  GOTO(:UOVNERR)
1992          DO(MD[-1])
1993          B-1
1994          GOTOR(MC)

1995 UTAIL(3)
1996          S=G
1997          G=S
1998          GOTOR(MC[-1])

1999 UHEAD(3)
2000          S=E
2001          G=S
2002          GOTOR(MC[-1])
```

to decompose a real into head and tail (two integers)

tail of F

back to tail of F, but now head will be supplied with 0 ?

S=M[57] head of F

deliver in tail of F

```
 0 UINVARITHMSEG(1)
 1          UINVARITHMSEG

 2 UINVENTIER(17)
 3          A = 1
 4          S = 4
 5          SE18
 6          S = MA[- 6]
 7      U,  S - 13, Z
 8      N,  SUBCD(:RSE19)
 9          F = :MA[- 11]
10          SUBCD(:RSE20)
11          S = 16
12          SE1
13          DOS(MA[- 11])
14          DO(ERV)
15          SUBC(:UENTIER)
16          S = UENTIERPRICE
17          GBILL + S, R
18          SE115
19          SE71

20 UINVSQRT(17)
21          A = 1
22          S = 4
23          SE18
24          S = MA[- 6]
25      U,  S - 13, Z
26      N,  SUBCD(:RSE19)
27          F = :MA[- 11]
28          SUBCD(:RSE20)
29          S = 16
30          SE1
31          DOS(MA[- 11])
32          DO(ERV)
33          SUBC(:USQRT)
34          S = USQRTPRICE
35          GBILL + S, R
36          SE115
37          SE71

38 UINVABS(18)
39          A = 1
40          S = 4
41          SE18
42          S = MA[- 6]
43      U,  S - 13, Z
44      N,  SUBCD(:RSE19)
45          F = :MA[- 11]
46          SUBCD(:RSE20)
47          S = 16
48          SE1
49          DOS(MA[- 11])
50          DO(ERV)
51          F + 0, R
52      N,  F = - F
53          S = UABSPRICE
54          GBILL + S, R
55          SE115
```

This is the beginning of segment filling.

Contains library procedures and the ALGOL-translator which functions like a library procedure
(All code reentrant)

arithmetic procedures

```
56          SE71

57 UINVLN(17)
58          A = 1
59          S = 4
60          SE18
61          S = MA[- 6]
62      U,  S - 13, Z
63      N,  SUBCD(:RSE19)
64          F = :MA[- 11]
65          SUBCD(:RSE20)
66          S = 16
67          SE1
68          DOS(MA[- 11])
69          DO(ERV)
70          SUBC(:ULN)
71          S = ULNPRICE
72          GBILL + S, F
73          SE115
74          SE71

75 UINVEXP(17)
76          A = 1
77          S = 4
78          SE18
79          S = MA[- 6]
80      U,  S - 13, Z
81      N,  SUBCD(:RSE19)
82          F = :MA[- 11]
83          SUBCD(:RSE20)
84          S = 16
85          SE1
86          DOS(MA[- 11])
87          DO(ERV)
88          SUBC(:UEXP)
89          S = UEXPRICE
90          GBILL + S, F
91          SE115
92          SE71

93 UINVCOS(17)
94          A = 1
95          S = 4
96          SE18
97          S = MA[- 6]
98      U,  S - 13, Z
99      N,  SUBCD(:RSE19)
100         F = :MA[- 11]
101         SUBCD(:RSE20)
102         S = 16
103         SE1
104         DOS(MA[- 11])
105         DO(ERV)
106         SUBC(:UCOS)
107         S = UCOSPRICE
108         GBILL + S, F
109         SE115
110         SE71

111 UINVSIN(17)
```

```
112           A = 1
113           S = 4
114           SE18
115           S = MA[- 6]
116      U,   S - 13, Z
117      N,   SUBCD(:PSE19)
118           F = :MA[- 11]
119           SUBCD(:PSE20)
120           S = 16
121           SE1
122           DOS(MA[- 11])
123           DO(ERV)
124           SUBC(:USIN)
125           S = USINPRICE
126           GBILL + S, P
127           SE115
128           SE71

129 UINVARCTAN(17)
130           A = 1
131           S = 4
132           SE18
133           S = MA[- 6]
134      U,   S - 13, Z
135      N,   SUBCD(:PSE19)
136           F = :MA[- 11]
137           SUBCD(:PSE20)
138           S = 16
139           SE1
140           DOS(MA[- 11])
141           DO(ERV)
142           SUBC(:UARCTAN)
143           S = UARCTANPRICE
144           GBILL + S, P
145           SE115
146           SE71

147 UINVSIGN(19)
148           A = 1
149           S = 4
150           SE18
151           S = MA[- 6]
152      U,   S - 13, Z
153      N,   SUBCD(:PSE19)
154           F = :MA[- 11]
155           SUBCD(:PSE20)
156           S = 16
157           SE1
158           DOS(MA[- 11])
159           DO(ERV)
160           F + 0, Z
161      N,   F = 1, E
162      N,   F = - 1
163           S = USIGNPRICE
164           GBILL + S, P
165           SE115
166           SE71

167 UINVMOD(19)
168           A = 1
```

```
169           S = 4
170           SE18
171           S = MA[- 6]
172       U,  S - 13, Z
173       N,  SUBCD(:RSE19)
174           F = :MA[- 11]
175           SUBCD(:RSE30)
176           S = 18
177           SE1
178           F = 0
179           MC = F
180           DOS(MA[- 11])
181           DO(RCV)
182           SUBC(:UMOD)
183           S = UMODPRICE
184           GBILL + S, P
185           SE115
186           SE71

187 UINVRE(17)                re(z)
188           A = 1
189           S = 4
190           SE18
191           S = MA[- 6]
192       U,  S - 13, Z
193       N,  SUBCD(:RSE19)
194           F = :MA[- 11]
195           SUBCD(:RSE30)
196           S = 18
197           SE1
198           B + 2
199           DOS(MA[- 11])
200           DO(RCV)
201           S = UREPRICE
202           GBILL + S, P
203           SE115
204           SE71

205 UINVIM(19)                im(z)
206           A = 1
207           S = 4
208           SE18
209           S = MA[- 6]
210       U,  S - 13, Z
211       N,  SUBCD(:RSE19)
212           F = :MA[- 11]
213           SUBCD(:RSE30)
214           S = 18
215           SE1
216           F = 0
217           MC = F
218           DOS(MA[- 11])
219           DO(RCV)
220           F = MC[- 2]
221           S = UIMPRICE
222           GBILL + S, P
223           SE115
224           SE71

225 UINVCOM(23)
```

```
226           A = 1
227           S = 4
228           SE18
229           S = MA[- 6]
230       W,  S - 17, Z
231       N,  SUBCD(:RSE19)
232           E = :MA[- 15]
233           SUBCD(:RSE20)
234           E = :MA[- 11]
235           SUBCD(:RSE20)
236           S = 16
237           SE1
238           DOS(MA[- 15])
239           DO(ERV)
240           MA[- 17] = E
241           DOS(MA[- 11])
242           DO(ERV)
243           MA[- 19] = E
244           E = MA[- 17]
245           S = UCOMPRICE
246           GBILL + S, E
247           SE115
248           SE71
249   UE244(7)
250           '002 004 004'
251           '130 135 021'
252           '016 025 022'
253           '027 020 135'
254           '015 030 030'
255           '033 135 027'
256           '036 025 776'
257   UE245(11)
258           '002 004 005'
259           '130 135 015'
260           '016 025 016'
261           '033 135 027'
262           '022 016 035'
263           '135 020 016'
264           '021 016 016'
265           '025 135 022'
266           '027 135 021'
267           '016 025 022'
268           '027 020 776'
269   UE246(12)
270           '002 004 006'
271           '130 135 015'
272           '016 016 025'
273           '035 012 025'
274           '135 027 022'
275           '016 035 135'
276           '020 016 021'
277           '016 016 025'
278           '135 022 027'
279           '135 021 016'
280           '025 022 027'
281           '020 776 000'
282   UE247(19)
283           '002 004 007'
284           '130 135 016'
285           '041 031 030'
```

com (a,b) delivers the complex value $a + \underline{\cdot} b$

dynamic

```
286           '027 016 027'
287           '035 135 027'
288           '022 016 035'
289           '135 020 016'
290           '021 016 016'
291           '025 135 022'
292           '027 135 014'
293           '030 026 031'
294           '025 016 041'
295           '016 135 026'
296           '012 014 021'
297           '035 034 037'
298           '016 033 021'
299           '016 017 017'
300           '022 027 020'
301           '776 000 000'

302 CENDRLDOC(30)
303           A = 1
304           S = 4
305           SE18
306           S = 9
307           S - MA[- 6], Z
308       N,  SUBCD(:RSE19)
309           UDEAF
310           G = URLSTREAMS
311           G = MG[0]
312           UBLOCKSTR = G
313           A = MG[6], R
314       N,  JUMP(12)
315           S = MT[16]
316           MG[17] + S
317           SUBCD(:UFILLRLSEG)
318           SUBCD(:USH4)
319           SUBCD(:USH8)
320           S = UBLOCKSTR
321           F = URLOWIDTH
322           MS[19] = -F
323           F = - 0
324           MS[21] = G
325           UTREXVWIT
326           SUBCD(:UMORTAL)
327           DO(MD[- 1])
328           S = CENDRLDOCPRICE
329           GBILL + S, R
330           SE115
331           SE71
332           '100 000 000'

333 UINVCOMPOSE(26)
334           A = 1
335           S = 4
336           SE18
337           S = MA[- 6]
338           S - 17, Z
339       N,  SUBCD(:RSE19)
340           F = :MA[- 15]
341           SUBCD(:RSE20)
342           F = :MA[- 11]
343           SUBCD(:RSE20)
```

end of plotterdocument

```
344           S = 17
345           SE1
346           DOS(MA[- 15])
347           DO(ERV)
348           S = E, Z
349       N,  SUBCD(:RSE88)
350           MC = G
351           DOS(MA[- 11])
352           DO(ERV)
353           S = E, Z
354       N,  SUBCD(:RSE88)
355           SUBC(:UCOMPOSE)
356           S = UCOMPOSERRICE
357           GBILL + S, P
358           SE115
359           SE71

360 UINVMEAD(17)
361           A = 1
362           S = 4
363           SE18
364           S = MA[- 6]
365           S - 13, Z
366       N,  SUBCD(:RSE19)
367           F = :MA[- 11]
368           SUBCD(:RSE20)
369           S = 16
370           SE1
371           DOS(MA[- 11])
372           DO(ERV)
373           SUBC(:UHEAD)
374           S = UHEADRRICE
375           GBILL + S, P
376           SE115
377           SE71
378 UINVTAIL(17)
379           A = 1
380           S = 4
381           SE18
382           S = MA[- 6]
383           S - 13, Z
384       N,  SUBCD(:RSE19)
385           F = :MA[- 11]
386           SUBCD(:RSE20)
387           S = 16
388           SE1
389           DOS(MA[- 11])
390           DO(ERV)
391           SUBC(:UTAIL)
392           S = UTAILRRICE
393           GBILL + S, P
394           SE115
395           SE71

396 UINVBIT(26)
397           A = 1
398           S = 4
399           SE18
400           S = MA[- 6]
401           S - 17, Z
```

```
402      N,  SUBCD(:RSE19)
403          F = :MA[- 15]
404          SUBCD(:RSE20)
405          F = :MA[- 11]
406          SUBCD(:RSE20)
407          $ = 17
408          SE1
409          DO$(MA[- 15])
410          DO(FRV)
411          $ = F, Z
412      N,  SUBCD(:RSE88)
413          MC = G
414          DOS(MA[- 11])
415          DO(FRV)
416          $ = F, Z
417      N,  SUBCD(:RSE88)
418          SUBC(:UBIT)
419          $ = UBITPRICE
420          GBILL + S, P
421          SE115
422          SE71

423 UINVSHIFT(33)
424          A = 1
425          $ = 4
426          SE18
427          $ = MA[- 6]
428          $ - 21, Z
429      N,  SUBCD(:RSE19)
430          F = :MA[- 19]
431          SUBCD(:RSE20)
432          F = :MA[- 15]
433          SUBCD(:RSE20)
434          F = :MA[- 11]
435          SUBCD(:RSE20)
436          $ = 18
437          SE1
438          DOS(MA[- 19])
439          DO(FRV)
440          $ = F, Z
441      N,  SUBCD(:RSE88)
442          MC = G
443          DOS(MA[- 15])
444          DO(FRV)
445          $ = F, Z
446      N,  SUBCD(:RSE88)
447          MC = G
448          DOS(MA[- 11])
449          DO(FRV)
450          $ = F, Z
451      N,  SUBCD(:RSE88)
452          SUBC(:USHIFT)
453          $ = USHIFTPRICE
454          GBILL + S, P
455          SE115
456          SE71

457 UINVOR(26)
458          A = 1
459          $ = 4
```

```
460           SE18
461           $ = MA[- 6]
462           $ - 17, Z
463      N,   SUBCD(:PSE19)
464           F = :MA[- 15]
465           SUBCD(:PSE20)
466           F = :MA[- 11]
467           SUBCD(:PSE20)
468           $ = 17
469           SE1
470           DOS(MA[- 15])
471           DO(ERV)
472           $ = F, Z
473      N,   SUBCD(:PSE88)
474           MC = G
475           DOS(MA[- 11])
476           DO(ERV)
477           $ = F, Z
478      N,   SUBCD(:PSE88)
479           SUBC(:UOR)
480           $ = UORPRICE
481           GBILL + $, P
482           SE115
483           SE71

484 UINVAND(26)
485           A = 1
486           $ = 4
487           SE18
488           $ = MA[- 6]
489           $ - 17, Z
490      N,   SUBCD(:PSE19)
491           F = :MA[- 15]
492           SUBCD(:PSE20)
493           F = :MA[- 11]
494           SUBCD(:PSE20)
495           $ = 17
496           SE1
497           DOS(MA[- 15])
498           DO(ERV)
499           $ = F, Z
500      N,   SUBCD(:PSE88)
501           MC = G
502           DOS(MA[- 11])
503           DO(ERV)
504           $ = F, Z
505      N,   SUBCD(:PSE88)
506           SUBC(:UAND)
507           $ = UANDRR:CE
508           GBILL + $, P
509           SE115
510           SE71

511 UINVPLOT(397)
512           A = 1
513           $ = 5
514           SE18
515           $ = MA[- 6]
516           $ - 21, Z
517      N,   SUBCD(:PSE19)
```

Many of these procedures have already been encountered as subroutines.

Translator compiles a simple subroutine call wherever permitted They do have to be present ~~for~~ as full fledged procedures in case they are handed over as actuals in the place of a formal procedure.

T plotter procedures, forget about it.

```
518          F = :RX[UA1]
519          SUBCD(:RSE20)
520          F = :RY[UA1]
521          SUBCD(:RSE20)
522          F = :RIREN[UA1]
523          SUBCD(:RSE20)
524          S = 16
525          SE1
526          DOS(RX[UA1])
527          DO(ERV)
528          RX[UA1] = F
529          DOS(RY[UA1])
530          DO(ERV)
531          RY[UA1] = F
532          DOS(RIREN[UA1])
533          DO(ERV)
534          S = F, Z
535      N,  SUBCD(:RSE88)
536          RIREN[UA1] = G
537          JUMP(27)
538          JUMP(S)                          " SWITCH
539          JUMP(87)                         " IREN0
540          JUMP(88)                         " IREN1
541          JUMP(92)                         " IREN2
542          JUMP(95)                         " IREN3
543          JUMP(196)                        " IREN4
544          JUMP(207)                        " IREN5
545          JUMP(213)                        " IREN6
546          JUMP(223)                        " IREN7
547          JUMP(227)                        " IREN8
548          JUMP(231)                        " IREN9
549          JUMP(232)                        " IREN10
550          JUMP(233)                        " IREN11
551          JUMP(234)                        " IREN12
552          JUMP(235)                        " IREN13
553          JUMP(236)                        " IREN14
554          JUMP(237)                        " IREN15
555          JUMP(238)                        " IREN16
556          JUMP(239)                        " IREN17
557          JUMP(242)                        " IREN18
558          JUMP(245)                        " IREN19
559          JUMP(248)                        " IREN20
560          JUMP(269)
561          JUMP(279)
562          JUMP(284)
563          JUMP(323)
564          JUMP(332)
565          G = URLSTREAMS
566          G = MG[0]
567          UBLOCKSTR = G
568          S = MG[6], P                     " DOCUMENT UNDER CONSTRUCTION?
569      Y,  JUMP(2)
570          CMEEN
571          UINVBEGINRLDOC
572          S = UFIRSTENTRANCE , P
573      Y,  UFIRSTENTRANCE = - B
574      Y,  USTATE1 = - B
575      Y,  USTATE2 = - B
576      Y,  USTATE4 = - B
```

```
577     Y,  USTATE8 = - B
578     Y,  USTATE16 = - B
579     Y,  USTATE32 = - B
580     Y,  USTATE64 = - B
581         $ = 16
582         SE0
583         $ = 16
584         SE1
585         $ = PIREN, P                    " ABSIPEN := ABS(IREN)
586     N,  $ = - $
587         PABSIPEN[UA2] = $
588     U,  $ - 19, P                       " ABSIPEN ≥ 20?
589     N,  $ + 0, Z                        " ABSIPEN = 0?
590     N,  JUMP(3)
591     U,  $ - 25, P                       " ABSIPEN > 25?
592     Y,  $ = 0
593         JUMP(- 56)                      " GOTO SWITCH
594         $ = USTATE1, P

595     Y,  $ = USTATE2, P
596     Y,  $ = USTATE4, P
597     Y,  $ = USTATE8, P
598     Y,  $ = USTATE16, P
599     N,  A = 239
600     N,  SUBCD(:UDYNERR)                 " STOP 1 - 26
601         G = - PIREN, Z
602     N,  F + 7, E                        " IREN ≤ 0 OF IREN > 6?
603     Y,  JUMP(10)
604         F = PX
605         F - PXMIN
606         F/PSCX
607         $ = F, Z
608     N,  SUBCD(:PSE88)
609         PLX[UA2] = G
610         F = PY
611         F - PYMIN
612         F/PSCY
613         JUMP(8)
614         $ = PABSIREN[UA2]
615     U,  $ - 8, P
616     Y,  JUMP(- 79)                      " GOTO SWITCH
617         F = PX
618         $ = F, Z
619     N,  SUBCD(:PSE88)
620         PLX[UA2] = G
621         F = PY
622         $ = F, Z
623     N,  SUBCD(:PSE88)
624         PLY[UA2] = G
625         $ = PABSIREN[UA2]
626         JUMP(- 89)                      " GOTO SWITCH
627         A = 238                         " IREN0
628         SUBCD(:UDYNERR)                 " STOP 1 - 25
629         $ = PREN, P                     " IREN1
630     Y,  A = 7
631     Y,  SUBCD(:UMOVE)
632     Y,  PREN = - B

633         JUMP(4)
634         $ = - PREN, P                   " IREN2
```

```
635     Y,  A = 3
636     Y,  $UBCD(:UMOVE)
637     Y,  PREN = B
638         $ = - PLX[UA2]                  " :PEN3
639         $ - 0, P
640     N,  $ = - PLY[UA2]
641     N,  $ - 0, P
642         G = PLY[UA2]
643         G - PYMAX
644     N,  F - 0, P
645     Y,  JUMP(6)
646         G = PLX[UA2]
647         G - PXMAX
648         F - 0, P
649     Y,  $ = - PABS:PEN[UA2]
650     Y,  $ + 4, P
651     N,  JUMP(2)
652         A = 240
653         $UBCD(:UDYNERR)             " STOP 1 - 27
654         G = PLX[UA2]
655         G - PXLAST
656         $ = F, Z
657     N,  $UBCD(:PSE88)
658         PDX[UA2] = G
659         G = PLY[UA2]
660         G - PYLAST
661         $ = F, Z
662     N,  $UBCD(:PSE88)
663         PDY[UA2] = G
664         G = PLX[UA2]
665         PXLAST = G
666         G = PLY[UA2]
667         PYLAST = G
668         $ = PDX[UA2]
669         $ - 0
670         PRIGHT[UA2] = $, P

671     N,  PDX[UA2] = - $, Z
672     N,  $ = 2
673     Y,  $ = 1
674         PLX[UA2] = $
675         $ = - PDY[UA2]
676         $ - 0, P
677     Y,  PDY[UA2] = $
678     Y,  $ = 8
679     N,  $ = 4
680         PLY[UA2] = $
681         $ + PLX[UA2]
682         PXY[UA2] = $
683         G = PDY[UA2]
684         G - PDX[UA2]
685         F - 0, P
686     Y,  $ = PLY[UA2]
687     Y,  PLX[UA2] = $
688     Y,  $ = PDX[UA2]
689     Y,  PI[UA2] = $
690     Y,  $ = PDY[UA2]
691     Y,  PDX[UA2] = $
692     Y,  $ = PI[UA2]
693     Y,  PDY[UA2] = $
```

```
694          G = RDX[UA2]
695          F + F
696          $ = F, Z
697      N,  SUBCR(:RSE88)
698          RDX2[UA2] = G
699          G = RDY[UA2]
700          F + F
701          $ = F, Z
702      N,  SUBCR(:RSE88)
703          RDY2[UA2] = G
704          $ = 0
705          RDEV[UA2] = $
706          RI[UA2] = S
707          JUMP(20)
708          G = RDEV[UA2]

709          G + RDY2[UA2]
710          $ = F, Z
711      N,  SUBCR(:RSE88)
712          RDEV[UA2] = G
713          G - RDX[UA2], P
714      Y,  JUMP(6)
715          F + 0, Z
716      Y,  S = RRIGHT[UA2], P
717      Y,  JUMP(3)
718          A = RLX[UA2]
719          SUBCR(:UMOVE)
720          JUMP(7)
721          A = RXY[UA2]
722          SUBCR(:UMOVE)
723          G = RDEV[UA2]
724          G - RDX2[UA2]
725          $ = F, Z
726      N,  SUBCR(:RSE88)
727          RDEV[UA2] = G
728          $ = RRLOTPRICE1
729          GBILL + S, R
730          SE115
731          $ = 1
732          RLUSS(RI[UA21)
733          $ - RDX[UA2], R
734      N,  JUMP(- 27)
735          $ = RABSIREN[UA2]
736          $ - 4, Z
737      Y,  RXLAST = S
738      Y,  RYLAST = S
739          JUMP(161)                    " GOTO END
740          $ = - RIREN, R               " IREN4
741      Y,  S = - RLX[UA2]
742      Y,  S - 0, R
743      N,  JUMP(- 110)                  " GOTO IREN2
744          G = RXMAX
745          F + 3000
746          $ = F, Z

747      N,  SUBCR(:RSE88)
748          RLX[UA2] = G
749          S = 0
750          RLY[UA2] = S
751          JUMP(- 118)                  " GOTO IREN2
```

```
752         G = RLX[UA2]                    " !REN5
753         RRLOT = F
754         S = - RLX[UA2]                  " RLOT51
755         S - 0, R
756         G = RLX[UA2]
757         G - RXMAX
758         JUMP(6)
759         G = RLY[UA2]                    " !REN6
760         RRLOT = F
761         S = - RLY[UA2]                  " RLOT61
762         S - 0, R
763         G = RLY[UA2]
764         G - RYMAX
765     N,  F - 0, R
766     N,  JUMP(136)                       " GOTO ENDD
767         A = 240                         " STOP 1 - 27
768         SUBCD(:UDYNERR)
769         JUMP(133)                       " GOTO ENDD
770         G = RLX[UA2]                    " !REN7
771         F * RSCX
772         F + RXMIN
773         RRLOT = F
774         JUMP(- 21)                      " GOTO RLOT51
775         G = RLY[UA2]                    " !REN8
776         F * RSCY
777         F + RYMIN
778         RRLOT = F
779         JUMP(- 19)                      " GOTO RLOT61
780         F = RXMIN                       " !REN9
781         JUMP(120)
782         F = RYMIN                       " !REN10
783         JUMP(118)
784         F = RSCX                        " !REN11
785         JUMP(116)
786         F = RSCY                        " !REN12
787         JUMP(114)
788         G = RXMAX                       " !REN13
789         JUMP(112)
790         G = RYMAX                       " !REN14
791         JUMP(110)
792         G = RXLAST                      " !REN15
793         JUMP(108)
794         G = RYLAST                      " !REN16
795         JUMP(106)
796         G = RXLAST                      " !REN17
797         F * RSCX
798         F + RXMIN
799         JUMP(102)
800         G = RYLAST                      " !REN18
801         F * RSCY
802         F + RYMIN
803         JUMP(98)
804         G = RREN, R                     " !REN19
805     Y,  F = 1
806     N,  F = - 1
807         JUMP(94)
808         F = RX                          " !REN20
809         S = F, Z
810     N,  $UBCD(:RSE88)
```

```
811          RLX[UA21 = G
812          A = G
813     U,   A - 3 , Z
814     Y,   A = 0
815     U,   A - 7 , Z
816     Y,   A = 0
817     U,   A - 16, Z
818     Y,   A = 3
819     U,   A - 32, Z
820     Y,   A = 7
821          A - 0, P
822     Y,   S = - A
823     Y,   S + 11, P
824     Y,   SUBCD(:UMOVE)
825     Y,   USTATE8 = - B
826     Y,   USTATE16 = - B

827     Y,   JUMP(73)                        " GOTO END
828          A = 237
829          SUBCD(:UDYNERR)
830          F = PX                          " IREN21
831          PXMIN = F
832          F = PY
833          PYMIN = F
834          USTATE1 = B
835          G = PIREN, P
836     Y,   USTATE16 = B
837     Y,   PREN = B
838     Y,   A = 3
839     Y,   SUBCD(:UMOVE)
840          JUMP(60)                        " GOTO END
841          F = PX                          " IREN22
842          PSCX = F
843          F = PY
844          PSCY = F
845          USTATE2 = B
846          JUMP(54)                        " GOTO END
847          F = PX                          " IREN23
848          S = F, Z
849     N,   SUBCD(:PSE88)
850          PXMAX = G
851          F = PY
852          S = F, Z
853     N,   SUBCD(:PSE88)
854          PYMAX = G
855          USTATE4 = B
856          G = PIREN, P
857     N,   JUMP(43)                        " GOTO END
858          USTATE8 = B
859          S = 0
860          PXLAST = S
861          PYLAST = S
862          S = PPLOTPR!CE2
863          GBILL * S, P
864          SE115

865          G = UBLOCKSTR
866          S = MG[20], Z                   " EDGE?
867     N,   A = 8
868     N,   SUBCD(:UMOVE)
```

```
869     N,  JUMP(- 8)
870         G = - RYMAX
871         F + URLOVIDTH
872         MC = F
873         F = 2
874         SUBC(:UDIDI)
875         RI[UA2] = G
876         F - 0, P.
877     N,  JUMP(8)
878         A = 4
879         SUBCD(:UMOVE)
880         S = RRLOTRRICE3
881         GBILL + S, P
882         SE115
883         S = 1
884         RI[UA2] - S, P
885     Y,  JUMP(- 8)
886         JUMP(14)                        " GOTO END
887         F = RX                          " IREN24
888         S = F, Z
889     N,  SUBCD(:RSE88)
890         RXLAST = G
891         F = RY
892         S = F, Z
893     N,  SUBCD(:RSE88)
894         RYLAST = G
895         USTATE8 = B
896         JUMP(4)                         " GOTO END
897         F = RX                          " IREN25
898         F - 0
899         RREN = G
900         USTATE16 = B
901         F = 0                           " END
902         RRLOT = F

903         SE6                             " ENDD
904         S = RRLOTRRICE
905         GBILL + S, P
906         SE115
907         F = RRLOT[UA1]
908         SE71

909 UINVSESEG3(1)
910         UINVSESEG3
911 UINVSESEG4(1)
912         UINVSESEG4
913 UINVSESEG5(1)
914         UINVSESEG5
915 UINVSESEG6(1)
916         UINVSESEG6
917 UINVSESEG7(1)
918         UINVSESEG7

919 UINVRLOTFRAME(102)
920         A = 1
921         S = 5
922         SE18
923         S = MA[- 6]
924         S - 33, Z
925     N,  SUBCD(:RSE19)
```

```
926           F = :REXMIN[UA1]
927           SUBCD(:RSE20)
928           F = :REYMIN[UA1]
929           SUBCD(:RSE20)
930           F = :REXMAX[UA1]
931           SUBCD(:RSE20)
932           F = :REYMAX[UA1]
933           SUBCD(:RSE20)
934           F = :RELXMAX[UA1]
935           SUBCD(:RSE20)
936           F = :RELYMAX[UA1]
937           SUBCD(:RSE20)
938           $ = 23
939           SE1
940           DOS(REXMIN[UA1])
941           DO(ERV)
942           REXMIN[UA1] = F
943           DOS(REYMIN[UA1])
944           DO(ERV)
945           REYMIN[UA1] = F
946           DOS(REXMAX[UA1])
947           DO(ERV)
948           REXMAX[UA1] = F
949           DOS(REYMAX[UA1])
950           DO(ERV)
951           REYMAX[UA1] = F
952           DOS(RELXMAX[UA1])
953           DO(ERV)
954           $ = F, Z
955       N,  SUBCD(:RSE88)
956           RELXMAX[UA1] = G
957           DOS(RELYMAX[UA1])
958           DO(ERV)
959           $ = F, Z
960       N,  SUBCD(:RSE88)
961           RELYMAX[UA1] = G
962           G = URLSTREAMS
963           G = MG[0]
964           UBLOCKSTR = G
965           $ = MG[6], R
966       Y,  JUMP(2)
967           CHEEN
968           UINVBEGINRLDOC
969           G = RELYMAX[UA1]
970           F - URLOWIDTHMIN
971           F - 0, R
972       Y,  A = 242
973       Y,  SUBCD(:UDYNERR)                " STOP 1 - 30
974           R + 2
975           $ = :REXMIN[UA1]
976           SUBCD(:RSE77)
977           $ = :REYMIN[UA1]
978           SUBCD(:RSE77)
979           DOS(RSE76)
980           + 0
981           + 21
982           S = 21
983           SUBCD(:RSE73)
984           UINVRLOT
985           R + 2
```

```
 986          DOS(R82E3)
 987          + 130
 988          UINVRLOTERAME[76]
 989          DOS(R82E3)
 990          + 130
 991          UINVRLOTERAME[80]
 992          DOS(RSE76)
 993          + 0
 994          + 22
 995          JUMP(8)
 996          E = REXMAX
 997          E - REXMIN
 998          G/RELXMAX
 999          SUBCD(:RSE72)
1000          E = REYMAX
1001          E - REYMIN
1002          G/RELYMAX
1003          SUBCD(:RSE72)
1004          S = 21
1005          SUBCD(:RSE73)
1006          UINVRLOT
1007          B + 2
1008          S = :RELXMAX[UA1]
1009          SUBC(:RSE78)
1010          S = :RELYMAX[UA1]
1011          SUBC(:RSE78)
1012          DOS(RSE76)
1013          + 0
1014          + 23
1015          S = 21
1016          SUBCD(:RSE73)
1017          UINVRLOT
1018          S = RRLOTERAMERRICE
1019          GBILL + S, R
1020          SE115
1021          SE71
1022 UFAILL(100)
1023          UF99
1024          UF200
1025          UF201
1026          UF202
1027          UF203
1028          UF204
1029          UF205
1030          UF206
1031          UF207
1032          UF208
1033          UF209
1034          UF210
1035          UF211
1036          UF212
1037          UF213
1038          UF214
1039          UF215
1040          UF216
1041          UF217
1042          UF218
1043          UF219
1044          UF220
1045          UF221
```

list of invariant addresses of dynamic error message-strings

1046	WF222
1047	WF223
1048	WF224
1049	WF225
1050	WF226
1051	WF227
1052	WF228
1053	WF229
1054	WF230
1055	WF231
1056	WF232
1057	WF233
1058	WF234
1059	WF235
1060	WF236
1061	WF237
1062	WF238
1063	WF239
1064	WF240
1065	WF241
1066	WF242
1067	WF243
1068	WF244
1069	WF245
1070	WF246
1071	WF247
1072	WF248
1073	WF249
1074	WF250
1075	WF251
1076	WF252
1077	WF253
1078	WF254
1079	WF255
1080	WF256
1081	WF257
1082	WF258
1083	WF259
1084	WF260
1085	WF261
1086	WF262
1087	WF263
1088	WF264
1089	WF265
1090	WF266
1091	WF267
1092	WF268
1093	0
1094	0
1095	0
1096	0
1097	0
1098	0
1099	0
1100	0
1101	0
1102	0
1103	0
1104	0
1105	0

```
1106          0
1107          0
1108          0
1109          0
1110          0
1111          0
1112          0
1113          0
1114          0
1115          0
1116          0
1117          0
1118          0
1119          0
1120          0
1121          0
1122          0
1123 WF248(15)
1124          '002 004 010'
1125          '130 135 033'
1126          '016 034 036'
1127          '025 035 012'
1128          '012 035 135'
1129          '037 012 027'
1130          '135 047 063'
1131          '061 064 063'
1132          '067 051 135'
1133          '035 016 024'
1134          '016 027 022'
1135          '027 014 030'
1136          '027 034 022'
1137          '034 035 016'
1138          '027 035 776'
1139 WF236(9)
1140          '002 003 006'
1141          '130 135 031'
1142          '016 027 135'
1143          '013 036 022'
1144          '035 016 027'
1145          '135 021 016'
1146          '035 135 031'
1147          '012 031 022'
1148          '016 033 776'
1149 WF237(9)
1150          '002 003 007'
1151          '130 135 030'
1152          '027 013 016'
1153          '024 016 027'
1154          '015 016 135'
1155          '031 025 030'
1156          '035 035 016'
1157          '033 014 030'
1158          '015 016 776'

1159 WF238(12)
1160          '002 003 010'
1161          '130 135 067'
1162          '070 063 064'
1163          '135 001 101'
1164          '002 005 135'
```

```
1165          '022 027 135'
1166          '031 025 030'
1167          '035 035 016'
1168          '033 031 033'
1169          '030 014 016'
1170          '015 036 033'
1171          '016 776 000'
1172 UE239(12)
1173          '002 003 011'
1174          '130 135 067'
1175          '070 063 064'
1176          '135 001 101'
1177          '002 006 135'
1178          '022 027 135'
1179          '031 025 030'
1180          '035 035 016'
1181          '033 031 033'
1182          '030 014 016'
1183          '015 036 033'
1184          '016 776 000'

1185 UE240(12)
1186          '002 004 000'
1187          '130 135 067'
1188          '070 063 064'
1189          '135 001 101'
1190          '002 007 135'
1191          '022 027 135'
1192          '031 025 030'
1193          '035 035 016'
1194          '033 031 033'
1195          '030 014 016'
1196          '015 036 033'
1197          '016 776 000'
1198 UE241(12)
1199          '002 004 001'
1200          '130 135 067'
1201          '070 063 064'
1202          '135 001 101'
1203          '002 010 135'
1204          '022 027 135'
1205          '031 025 030'
1206          '035 035 016'
1207          '033 031 033'
1208          '030 014 016'
1209          '015 036 033'
1210          '016 776 000'

1211 UE242(12)
1212          '002 004 002'
1213          '130 135 067'
1214          '070 063 064'
1215          '135 001 101'
1216          '003 000 135'
1217          '022 027 135'
1218          '031 025 030'
1219          '035 035 016'
1220          '033 031 033'
1221          '030 014 016'
1222          '015 036 033'
```

dynamic error message strings.

```
1223          '016 776 000'
1224 UF243(12)
1225          '002 004 003'
1226          '130 135 067'
1227          '070 063 064'
1228          '135 001 101'
1229          '003 001 135'
1230          '022 027 135'
1231          '031 025 030'
1232          '035 035 016'
1233          '033 031 033'
1234          '030 014 016'
1235          '015 036 033'
1236          '016 776 000'

1237 UINVPLOTCURVE(511)
1238          A = 1
1239          $ = 6
1240          SE18
1241          $ = MA[- 6]
1242          $ - 21, Z
1243      N,  SUBCD(:PSE19)
1244          F = :RCX[UA1]
1245          SUBCD(:PSE20)
1246          F = :RCY[UA1]
1247          SUBCD(:PSE20)
1248          F = :RCI[UA1]
1249          SUBCD(:PSE20)
1250          $ = 16
1251          SE1
1252          DOS(RCX[UA1])
1253          DO(ERV)
1254          RCX[UA1] = F
1255          DOS(RCY[UA1])
1256          DO(ERV)
1257          RCY[UA1] = F
1258          DOS(RCI[UA1])
1259          DO(ERV)
1260          $ = F, Z
1261      N,  SUBCD(:PSE88)
1262          RCI[UA1] = G
1263          G = URLSTREAMS
1264          G = MG[0]
1265          UBLOCKSTR = G
1266          $ = MG[6],P
1267      Y,  JUMP(2)
1268          CHEEN
1269          UINVBEGINRLDOC
1270          $ = 71
1271          SE0
1272          $ = 27
1273          SE1
1274          F = 0
1275          RCPLOTCURVE = F
1276          G = - RCI                    " INIT
1277          F + 2, P
1278      N,  JUMP(50)
1279          F - 2, P
1280      N,  JUMP(3)
1281          A = 241                      " OOPS
```

```
1282          SUBCD(:UDYNERR)                  " STOP 1 - 28
1283          JUMP(463)
1284          G = RCI, Z
1285      Y,  RCDERIV = B
1286      N,  RCDERIV = - B
1287          RCXISX = B
1288          F = 0
1289          RCCOUNT = G
1290          RCX1 = F
1291          RCX2 = F
1292          RCY1 = F
1293          RCY2 = F
1294          B + 2
1295          DOS(RSE76)
1296          0
1297          0
1298          DOS(RSE76)
1299          0
1300          0
1301          DOS(RSE76)
1302          0
1303          12
1304          S = 21
1305          SUBCD(:RSE73)
1306          UINVPLOT
1307          MC = F
1308          B + 2
1309          DOS(RSE76)
1310          0
1311          0
1312          DOS(RSE76)
1313          0
1314          0
1315          DOS(RSE76)
1316          0
1317          11
1318          S = 21

1319          SUBCD(:RSE73)
1320          UINVPLOT
1321          F/MC[- 2]
1322          RCSCALEF = F
1323          G = RCDERIV, R
1324      Y,  F = RCX
1325      Y,  F * RCSCALEF
1326      Y,  RCY0R[UA2] = F
1327          USTATE32 = B
1328          JUMP(418)
1329          G = RCI                          " CHECK
1330          F - 4, R
1331      N,  S = - USTATE32, R
1332      Y,  JUMP(- 52)
1333          G = - RCI
1334          F + 3, R
1335      N,  S = RCCOUNT
1336      N,  S - 4, Z
1337      N,  JUMP(- 57)
1338          F = RCX1                         " SHIFT
1339          RCX0[UA2] = F
```

more plot proc.

```
1340          F = RCY1
1341          RCY0[UA2] = F
1342          F = RCX2
1343          RCX1 = F
1344          F = RCY2
1345          RCY1 = F
1346          G = RC1                          " NEWPOINT
1347          F - 2, Z
1348      N,  JUMP(68)
1349          F = RCX
1350          MC = F
1351          B + 2
1352          DOS(RSE76)
1353          + 0
1354          + 0
1355          DOS(RSE76)
1356          + 0

1357          + 0
1358          DOS(RSE76)
1359          + 0
1360          + 9
1361          S = 21
1362          SUBCD(:RSE73)
1363          UINVPLOT
1364          F - MC[- 2]
1365          MC = - F
1366          B + 2
1367          DOS(RSE76)
1368          + 0
1369          + 0
1370          DOS(RSE76)
1371          + 0
1372          + 0
1373          DOS(RSE76)
1374          + 0
1375          + 11
1376          S = 21
1377          SUBCD(:RSE73)
1378          UINVPLOT
1379          MC = F
1380          F = MC[- 4]
1381          F/MC
1382          RCX2 = F
1383          F = RCY
1384          MC = F
1385          B + 2
1386          DOS(RSE76)
1387          + 0
1388          + 0
1389          DOS(RSE76)
1390          + 0
1391          + 0

1392          DOS(RSE76)
1393          + 0
1394          + 10
1395          S = 21
1396          SUBCD(:RSE73)
1397          UINVPLOT
```

```
1398          F - MC[- 2]
1399          MC = - F
1400          B + 2
1401          DOS(RSE76)
1402          + 0
1403          + 0
1404          DOS(RSE76)
1405          + 0
1406          + 0
1407          DOS(RSE76)
1408          + 0
1409          + 12
1410          S = 21
1411          SUBCD(:RSE73)
1412          UINVPLOT
1413          MC = F
1414          F = MC[- 4]
1415          F/MC
1416          RCY2 = F
1417          G = - RCCOUNT
1418          F + 3, R
1419     N,   JUMP(4)
1420          F - 4
1421          RCCOUNT = - G
1422          F + 3, Z
1423     N,   JUMP(323)
1424     N,   F = RCY1R
1425     N,   RCY0R[UA2] = F
1426          F = RCX0[UA2]                    " POSITION
1427          S = F, Z
1428     N,   SUBCD(:RSE88)
1429          RCX0[[UA2] = G

1430          F = RCY0[UA2]
1431          S = F, Z
1432     N,   SUBCD(:RSE88)
1433          RCY0[[UA2] = G
1434          F = UCLEAR
1435          MC = F
1436          DOS(RSE76)
1437          + 0
1438          + 0
1439          DOS(RSE76)
1440          + 0
1441          + 0
1442          DOS(RSE76)
1443          + 0
1444          + 15
1445          S = 21
1446          SUBCD(:RSE73)
1447          UINVPLOT
1448          G - RCX0[[UA2], Z
1449     N,   JUMP(16)
1450          F = UCLEAR
1451          MC = F
1452          DOS(RSE76)
1453          + 0
1454          + 0
1455          DOS(RSE76)
1456          + 0
```

```
1457          + 0
1458          DOS(RSE76)
1459          + 0
1460          + 16
1461          S = 21
1462          SUBCD(:RSE73)
1463          UINVRLOT
1464          G = RCY0![UA2], Z
1465      Y,  JUMP(12)
1466          F = UCLEAR
1467          MC = F

1468          S = :RCX0![UA2]
1469          SUBC(:RSE78)
1470          S = :RCY0![UA2]
1471          SUBC(:RSE78)
1472          DOS(RSE76)
1473          - 0
1474          - 2
1475          S = 21
1476          SUBCD(:RSE73)
1477          UINVRLOT
1478          F = UCLEAR                        " PENDOWN
1479          MC = F
1480          DOS(RSE76)
1481          + 0
1482          + 0
1483          DOS(RSE76)
1484          + 0
1485          + 0
1486          DOS(RSE76)
1487          + 0
1488          + 19
1489          S = 21
1490          SUBCD(:RSE73)
1491          UINVRLOT
1492          F = 0, R
1493      N,  JUMP(12)
1494          F = UCLEAR
1495          MC = F
1496          S = :RCX0![UA2]
1497          SUBC(:RSE78)
1498          S = :RCY0![UA2]
1499          SUBC(:RSE78)
1500          DOS(RSE76)
1501          - 0
1502          - 1
1503          S = 21
1504          SUBCD(:RSE73)
1505          UINVRLOT

1506          RCF![RSTTBY[UA2] = B
1507          G = RCX!SX, R                     " SCALE
1508      N,  JUMP(53)
1509          F = RCX1
1510          F = RCX0[UA2]
1511          RCXS1[UA2] = F
1512          F = RCY1
1513          F = RCY0[UA2]
1514          RCYS1[UA2] = F
```

```
1515          F = RCX2
1516          F - RCX0[UA2]
1517          RCXS2[UA2] = F
1518          F = RCY2
1519          F - RCY0[UA2]
1520          RCYS2[UA2] = F
1521          G = - RCX0[UA2]
1522          RCXSMIN[UA2] = G
1523          F = UCLEAR
1524          MC = F
1525          DOS(RSE76)
1526          + 0
1527          + 0
1528          DOS(RSE76)
1529          + 0
1530          + 0
1531          DOS(RSE76)
1532          + 0
1533          + 13
1534          S = 21
1535          SUBCD(:RSE73)
1536          UINVPLOT
1537          G - RCX0[UA2]
1538          S = F, Z
1539      N,  SUBCD(:RSE88)
1540          RCXSMAX[UA2] = G
1541          G = - RCY0[UA2]
1542          RCYSMIN[UA2] = G
1543          F = UCLEAR
1544          MC = F
1545          DOS(RSE76)
1546          + 0
1547          + 0
1548          DOS(RSE76)
1549          + 0
1550          + 0
1551          DOS(RSE76)
1552          + 0
1553          + 14
1554          S = 21
1555          SUBCD(:RSE73)
1556          UINVPLOT
1557          G - RCY0[UA2]
1558          S = F, Z
1559      N,  SUBCD(:RSE88)
1560          RCYSMAX[UA2] = G
1561          JUMP(52)
1562          F = RCY1
1563          F - RCY0[UA2]
1564          RCXS1[UA2] = F
1565          F = RCX1
1566          F - RCX0[UA2]
1567          RCYS1[UA2] = F
1568          F = RCY2
1569          F - RCY0[UA2]
1570          RCXS2[UA2] = F
1571          F = RCX2
1572          F - RCX0[UA2]
1573          RCYS2[UA2] = F
```

```
1574          G = - RCY0I[UA2]
1575          RCXSMIN[UA2] = G
1576          F = UCLEAR
1577          MC = F
1578          DOS(RSE76)
1579          + 0
1580          + 0
1581          DOS(RSE76)

1582          + 0
1583          + 0
1584          DOS(RSE76)
1585          + 0
1586          + 14
1587          S = 21
1588          SUBCD(:RSE73)
1589          UINVPLOT
1590          G - RCY0I[UA2]
1591          S = F, Z
1592      N,  SUBCD(:RSE88)
1593          RCXSMAX[UA2] = G
1594          G = - RCX0I[UA2]
1595          RCYSMIN[UA2] = G
1596          F = UCLEAR
1597          MC = F
1598          DOS(RSE76)
1599          + 0
1600          + 0
1601          DOS(RSE76)
1602          + 0
1603          + 0
1604          DOS(RSE76)
1605          + 0
1606          + 13
1607          S = 21
1608          SUBCD(:RSE73)
1609          UINVPLOT
1610          G - RCX0I[UA2]
1611          S = F, Z
1612      N,  SUBCD(:RSE88)
1613          RCYSMAX[UA2] = G
1614          G = RCEIRSTTRY[UA2], R
1615      N,  JUMP(129)
1616          G = RCI                          " SLOPE
1617          F - 3, Z
1618      N,  JUMP(21)
1619          F = RCXS1[UA2]

1620          F * RCYS1[UA2]
1621          F + F
1622          RCTEMP[UA2] = F
1623          F = RCYS1[UA2]
1624          F * F
1625          MC = F
1626          F = RCXS1[UA2]
1627          F * F
1628          F - MC[- 2]
1629          RCTEMP2[UA2] = - F
1630          F = RCTEMP[UA2]
1631          F * RCY0R[UA2]
```

```
1632          F - RCTEMP2[UA2]
1633          MC = F
1634          F = RCYOR[UA2]
1635          F * RCTEMP2[UA2]
1636          F + RCTEMP[UA2]
1637          F/MC[- 2]
1638          RCY1P = F
1639          JUMP(41)
1640          G = RCI
1641          F - 4, Z
1642      N,  JUMP(8)
1643          F = RCX
1644          F * RCSCALEF
1645          S = RCXISX, P
1646      N,  MC = F
1647      N,  F = 1
1648      N,  F/MC[- 2]
1649          RCY1P = F
1650          JUMP(30)
1651          F = RCXS2[UA2]
1652          F - RCXS1[UA2]
1653          F * RCXS2[UA2]
1654          MC = F
1655          F = RCYS2[UA2]
1656          F - RCYS1[UA2]
1657          F * RCYS2[UA2]

1658          F + MC[- 2]
1659          RCTEMP[UA2] = F
1660          F = RCYS1[UA2]
1661          F * F
1662          F * RCXS2[UA2]
1663          MC = F
1664          F = - RCYS1[UA2]
1665          F * RCYS2[UA2]
1666          F + RCTEMP[UA2]
1667          F * RCXS1[UA2]
1668          F + MC[- 2]
1669          MC = F
1670          F = RCXS1[UA2]
1671          F * F
1672          F * RCYS2[UA2]
1673          MC = F
1674          F = - RCXS1[UA2]
1675          F * RCXS2[UA2]
1676          F + RCTEMP[UA2]
1677          F * RCYS1[UA2]
1678          F + MC[- 2]
1679          F/MC[- 2]
1680          RCY1P = F
1681          G = RCCOUNT
1682          F - 3, Z
1683      Y,  S = 4
1684      Y,  RCCOUNT = S
1685      Y,  S = - RCDERIV, P
1686      N,  JUMP(29)
1687          F = RCXS1[UA2]                    " BEGINSLOPE
1688          F * F
1689          MC = F
1690          F = RCYS1[UA2]
```

```
1691            F * F
1692            F + MC[- 2]
1693            RCTEMP[UA2] = F
1694            F = RCXS2[UA2]
1695            F * F
1696            MC = F

1697            F = RCYS2[UA2]
1698            F * F
1699            F + MC[- 2]
1700            RCTEMP2[UA2] = F
1701            F = RCXS1[UA2]
1702            F * RCTEMP2[UA2]
1703            MC = F
1704            F = - RCXS2[UA2]
1705            F * RCTEMP[UA2]
1706            F + MC[- 2]
1707            MC = F
1708            F = RCYS1[UA2]
1709            F * RCTEMP2[UA2]
1710            MC = F
1711            F = - RCYS2[UA2]
1712            F * RCTEMP[UA2]
1713            F + MC[- 2]
1714            F/MC[- 2]
1715            RCYOP[UA2] = F
1716            F = RCYOP[UA2], P
1717      Y,    RCMAX1[UA2] = F
1718      N,    RCMAX1[UA2] = - F
1719            F = RCY1P, P
1720      Y,    RCTEMP[UA2] = F
1721      N,    RCTEMP[UA2] = - F
1722            F = RCMAX1[UA2]                  " INTERCHANGE
1723            F - RCTEMP[UA2]
1724            F - 0, P
1725            F = 1
1726      Y,    F/RCTEMP[UA2]
1727      N,    F/RCMAX1[UA2]
1728            RCMAX2[UA2] = F
1729      N,    F = RCTEMP[UA2]
1730      N,    RCMAX1[UA2] = F
1731            F = RCMAX1[UA2]
1732            F - RCMAX2[UA2]
1733            F - 0, P
1734      N,    JUMP(10)

1735            S = RCXISX
1736            RCXISX = - S
1737            F = 1
1738            F/RCYOP[UA2]
1739            RCYOP[UA2] = F
1740            F = 1
1741            F/RCY1P
1742            RCY1P = F
1743            RCFIRSTTRY[UA2] = - B
1744            JUMP(- 238)
1745            SE33
1746            UINVPLOTCURVE1           " POLYNOM
1747            SE33
1748            UINVPLOTCURVE1[381]      " EXIT
```

```
1749 UINVPLOTCURVE1(411)
1750          F = RCX1                          " POLYNOM
1751          F + MT[9]
1752          SUBC(:UENTIER)
1753          G - RCX0[UA2]
1754          RCTEMP[UA2] = F
1755          F = RCY1
1756          F + MT[4]
1757          SUBC(:UENTIER)
1758          G - RCY0[UA2]
1759          RCTEMP2[UA2] = F
1760          JUMP(2)
1761          '777600000'
1762          '000000001'
1763          F = RCTEMP[UA2]
1764          S = F, Z
1765      N,  SUBCD(:RSE88)
1766          A = G
1767          F = RCTEMP2
1768          S = F, Z
1769      N,  SUBCD(:RSE88)
1770          S = G
1771          G = RCXISX, R
1772    N,    RCSA(27)
1773          RCXS1[ = A
1774          RCYS1[ = S
1775          DO(MD[- 1])
1776          G = RCXS1[UA2], R
1777      Y,  S = 1
1778      N,  S = - 1
1779          F + 0, Z
1780      Y,  S = 0
1781          RCINCRX[UA2] = S
1782          F * F
1783          RCTEMP[UA2] = F
1784          G = - RCYS1[[UA2]
1785          F + F
1786          G/RCXS1[[UA2]

1787          F + RCY1R
1788          F + RCY0R[UA2]
1789          RCTEMP2[UA2] = F
1790          F/RCTEMP[UA2]
1791          RCA[UA2] = F
1792          F = - RCTEMP2[UA2]
1793          F - RCY0R[UA2]
1794          G * RCXS1[[UA2]
1795          G + RCYS1[[UA2]
1796          F/RCTEMP[UA2]
1797          RCB[UA2] = F
1798          F = 0
1799          RCLX[UA2] = G
1800          RCLY[UA2] = G
1801          RCF[UA2] = G
1802          S = RCXISX, R
1803      Y,  F = RCY1R
1804      N,  F = 1
1805      N,  F/RCY1R
1806          F/RCSCALEF
```

```
1807          RCRLOTCURVE = F
1808          G = RCXS1![UA2], F
1809      N,  F = - F
1810          RCABSXS1[UA2] = G, Z
1811      N,  JUMP(9)                          " SCALING
1812          S = RCCOUNT
1813          S - 3, Z
1814      Y,  S = 2
1815      Y,  RCCOUNT = S
1816          S = 1
1817          RC!SC6[UA2] = S
1818          JUMP(246)                        " GOTO ESCAPE
1819          - 1305041
1820          + 17562601
1821          G = RCABSXS1[UA2]
1822          F * 6
1823          SUBC(:ULN)

1824          F/MT[- 6]
1825          F + 1
1826          S = F, Z
1827      N,  SUBCD(:RSE88)
1828          RCLN1[UA2] = G
1829          G = RCABSXS1[UA2]
1830          F + 3
1831          F * F
1832          F * 3
1833          SUBC(:ULN)
1834          F/MT[- 16]
1835          F + 1
1836          S = F, Z
1837      N,  SUBCD(:RSE88)
1838          RCLN2[UA2] = G
1839          G = RCABSXS1[UA2]
1840          F + 5
1841          MC = F
1842          F * F
1843          F * MC[- 2]
1844          SUBC(:ULN)
1845          F/MT[- 27]
1846          F + 1
1847          S = F, Z
1848      N,  SUBCD(:RSE88)
1849          RCLN3[UA2] = G
1850          F = 2
1851          MC = F
1852          G = RCLN2[UA2]
1853          SUBC(:UTTR)
1854          RCSC1[UA2] = F
1855          F = 2
1856          MC = F
1857          G = RCLN3[UA2]
1858          SUBC(:UTTR)
1859          RCSC2[UA2] = F
1860          F = 2
1861          MC = F

1862          G = RCLN3[UA2]
1863          G - RCLN2[UA2]
1864          F - 1
```

```
1865          SUBC(:UTTR)
1866          S = F, Z
1867     N,   SUBCD(:RSE88)
1868          RC!SC1[UA2] = G
1869          F + F
1870          S = F, Z
1871     N,   SUBCD(:RSE88)
1872          RC!SC2[UA2] = G
1873          F = 2
1874          MC = F
1875          G = RCLN2[UA2]
1876          G - RCLN1[UA2]
1877          F - 1
1878          SUBC(:UTTR)
1879          S = F, Z
1880     N,   SUBCD(:RSE88)
1881          RC!SC3[UA2] = G
1882          F + F
1883          S = F, Z
1884     N,   SUBCD(:RSE88)
1885          RC!SC4[UA2] = G
1886          F = 2
1887          MC = F
1888          G = RCLN1[UA2]
1889          F - 1
1890          SUBC(:UTTR)
1891          S = F, Z
1892     N,   SUBCD(:RSE88)
1893          RC!SC5[UA2] = G
1894          F + F
1895          S = F, Z
1896     N,   SUBCD(:RSE88)
1897          RC!SC6[UA2] = G
1898          G + RC!SC5[UA2]
1899          S = F, Z

1900     N,   SUBCD(:RSE88)
1901          RC!SC7[UA2] = G
1902          F = - RCB[UA2]                    " DIFFERENCES
1903          G * RCINCRX[UA2]
1904          F + RCA[UA2]
1905          F + RCY08[UA2]
1906          F * RCSC1[UA2]
1907          G * RCINCRX[UA2]
1908          S = F, Z
1909     N,   SUBCD(:RSE88)
1910          RCDELTA[UA2] = G
1911          F = RCB[UA2]
1912          F + F
1913          MC = F
1914          F = - 6
1915          F + RCA[UA2]
1916          G * RCINCRX[UA2]
1917          F + MC[- 2]
1918          F * RCSC2[UA2]
1919          S = F, Z
1920     N,   SUBCD(:RSE88)
1921          RCDELTA2[UA2] = G
1922          F = 6
1923          F * RCA[UA2]
```

```
1924          F * RCSC2[UA2]
1925          G * RCINCRX[UA2]
1926          $ = F, Z
1927      N,  SUBCD(:RSE88)
1928          RCDELTA3[UA2] = G
1929          $ = RCXISX, P
1930      Y,  F = 1
1931      N,  F = 5
1932          G - RCINCRX[UA2]
1933          $ = F, Z
1934      N,  SUBCD(:RSE88)
1935          RCWHICH1[UA2] = G
1936          F + 8
1937          $ = F, Z

1938      N,  SUBCD(:RSE88)
1939          RCWHICH2[UA2] = G
1940          F + 8
1941          $ = F, Z
1942      N,  SUBCD(:RSE88)
1943          RCWHICH3[UA2] = G
1944          G = RCYSMIN[UA2]
1945          G * RCISC6[UA2]
1946          $ = F, Z
1947      N,  SUBCD(:RSE88)
1948          RCYSMIN[UA2] = G
1949          G = RCYSMAX[UA2]
1950          G * RCISC6[UA2]
1951          $ = F, Z
1952      N,  SUBCD(:RSE88)
1953          RCYSMAX[UA2] = G
1954          RCOKSOFAR[UA2] = B
1955          G = RCLX[UA2]                    " LOOP
1956          G + RCINCRX[UA2]
1957          $ = F, Z
1958      N,  SUBCD(:RSE88)
1959          RCLX[UA2] = G
1960          G = RCDELTA2[UA2]
1961          G + RCDELTA3[UA2]
1962          $ = F, Z
1963      N,  SUBCD(:RSE88)
1964          RCDELTA2[UA2] = G
1965          F = F, Z
1966      N,  F = 1, E
1967      N,  F = - 1
1968          G * RCISC1[UA2]
1969          G + RCDELTA2[UA2]
1970          MC = F
1971          G = RCISC2[UA2]
1972          SUBC(:UDIDI)
1973          G + RCDELTA[UA2]
1974          $ = F, Z
1975      N,  SUBCD(:RSE88)

1976          RCDELTA[UA2] = G
1977          F = F, Z
1978      N,  F = 1, E
1979      N,  F = - 1
1980          G * RCISC3[UA2]
1981          G + RCDELTA[UA2]
```

```
1982          MC = E
1983          G = RCISC4[UA2]
1984          SUBC(:UDIDI)
1985          G + RCF[UA2]
1986          S = E, Z
1987      N,  SUBCD(:RSE88)
1988          RCF[UA2] = G
1989          G = RCF[UA2]                    " LOOP2
1990          G - RCLY[UA2]
1991          S = E, Z
1992      N,  SUBCD(:RSE88)
1993          RCDIF[UA2] = G, P
1994      N,  E = - E
1995          RCABSDIF[UA2] = G
1996          G - RCISC5[UA2]
1997          E - 0
1998          RCSMALL[UA2] = - G
1999          G = RCABSDIF[UA2]
2000          G - RCISC7[UA2]
2001          E - 0
2002          RCHUGE[UA2] = G
2003          G = RCSMALL[UA2], P
2004      Y,  S = RCVHICH1[UA2]
2005      Y,  JUMP(3)
2006          G = RCHUGE[UA2], P
2007      Y,  S = RCVHICH3[UA2]
2008      N,  S = RCVHICH2[UA2]
2009          G = - RCDIF[UA2]
2010          E - 0, P
2011          S + :MT[125]
2012      Y,  A = MS[1]
2013      N,  A = MS[0]

2014          SUBCD(:UMOVE)
2015          S = RCSMALL[UA2], P
2016      Y,  JUMP(28)
2017          G = RCDIF[UA2], Z
2018      Y,  JUMP(7)
2019          E = E, P
2020          G = RCISC6[UA2]
2021      N,  E = - E
2022          G + RCLY[UA2]
2023          S = E, Z
2024      N,  SUBCD(:RSE88)
2025          RCLY[UA2] = G
2026          G = RCOKSOFAR[UA2], P
2027      N,  JUMP(11)
2028          G = - RCLY[UA2]
2029          G + RCYSMIN[UA2]
2030          E - 0, P
2031      Y,  JUMP(4)
2032          G = RCLY[UA2]
2033          G - RCYSMAX[UA2]
2034          E - 0, P
2035      N,  JUMP(3)
2036          A = 240                         " STOP 1 - 27
2037          SUBCD(:UDYNERR)
2038          RCOKSOFAR[UA2] = - B
2039          G = RCHUGE[UA2], P
2040      N,  JUMP(4)
```

```
2041          S = RCRLOTCURVERRICE1
2042          GBILL + S, R
2043          SE115
2044          JUMP(- 56)
2045          G = RCOKSOFAR[UA2], R              " TEST
2046      N,  JUMP(11)
2047          G = - RCLX[UA2]
2048          G + RCXSMIN[UA2]
2049          F - 0, R
2050      Y,  JUMP(4)
2051          G = RCLX[UA2]

2052          G - RCXSMAX[UA2]
2053          F - 0, R
2054      N,  JUMP(3)
2055          A = 240
2056          SUBCD(:UDYNERR)
2057          RCOKSOFAR[UA2] = - B
2058          G = RCLX[UA2]
2059          G - RCXS1[UA2], Z
2060      Y,  JUMP(4)
2061          S = RCRLOTCURVERRICE2
2062          GBILL + S, R
2063          SE115
2064          JUMP(- 110)
2065          G = RCXISX, R                      " ESCARE
2066          F = UCLEAR
2067          MC = F
2068      N,  JUMP(23)
2069          DOS(R82E3)
2070          + 129
2071          UINVRLOTCURVE1[329]
2072          DOS(R82E3)
2073          + 129
2074          UINVRLOTCURVE1[332]
2075          DOS(RSE76)
2076          + 0
2077          + 24
2078          JUMP(9)
2079          G = RCX01
2080          G + RCLX
2081          SUBCD(:RSE72)
2082          G = RCLY
2083          MC = G
2084          G = RC1SC6
2085          SUBC(:UE1D1)
2086          G + RCY01
2087          SUBCD(:RSE72)
2088          S = 21
2089          SUBCD(:RSE73)
2090          UINVRLOT

2091          JUMP(22)
2092          DOS(R82E3)
2093          + 129
2094          UINVRLOTCURVE1[352]
2095          DOS(R82E3)
2096          + 129
2097          UINVRLOTCURVE1[358]
2098          DOS(RSE76)
```

```
2099          + 0
2100          + 24
2101          JUMP(9)
2102          G = RCLY
2103          MC = G
2104          G = RCISC6
2105          SUBC(:UEIDI)
2106          G + RCX0;
2107          SUBCD(:RSE72)
2108          G = RCY0;
2109          G + RCLX
2110          SUBCD(:RSE72)
2111          S = 21
2112          SUBCD(:RSE73)
2113          UINVPLOT
2114          F = UCLEAR
2115          MC = F
2116          DOS(RSE76)
2117          - 0
2118          - 1
2119          DOS(RSE76)
2120          + 0
2121          + 0
2122          DOS(RSE76)
2123          + 0
2124          + 25
2125          S = 21
2126          SUBCD(:RSE73)
2127          UINVPLOT
2128          G = RCI
2129          F - 2, P
2130      Y,  USTATE32 = - B
2131          $E6                                " EXIT
2132          S = RCPLOTCURVEPRICE
2133          GBILL + S, P
2134          $E115
2135          F = RCPLOTCURVE[UA1]
2136          $E71
2137          1                                  " ACT
2138          1
2139          2
2140          2
2141          4
2142          4
2143          8
2144          8
2145          5
2146          9
2147          6
2148          10
2149          5
2150          6
2151          9
2152          10
2153          4
2154          8
2155          4
2156          8
2157          1
```

these plotterprocedures were handcoded from original ALGOL-text.

2158	2
2159	1
2160	2

C1063.3
Part 2

```
0 UINVARVALUE(1)
1         UINVARVALUE

2 UARVALUE(190)
3         GLFLOC = $                          " SAVE :F[0]
4         GLARCO = A                          " SAVE :ARRAYCOUPLE LOCAL ARRAY
5         GLTYPE = G                          " SAVE TYPE SPECIFICATION
6         A = MC[- 1]
7         GLRETURN = A                        " SAVE INVARIANT RETURNADDRESS
8         DO(MD[- 1])
9         $ = M$                              " :ARRAYCOUPLE ORIGIONAL ARRAY
10        $ = M$                              " :B[0] IN MAPPING TABLE
11        $ = M$[- 1]                         " DIMENSION
12        $ + :MS[10]                         " SPACE RESERVATION FOR BOUNDS
13        SE1                                 " AND MAPPING TABLE
14        $ = GLFLOC                          " :F[0]
15        $ = M$                              " :ARRAYCOUPLE
16        $ = M$                              " :B[0]
17        A = - MS[- 1]                       " DIMENSION NEGATIV
18        A + A
19        A + :MS[- 2]                        " :FIRST BOUND PAIR
20        G = MA
21        G + MA[1]
22        MC = G                              " STACK LOWERBOUND
23        G - MA
24        F - 1
25        MC = G                              " STACK UPPERBOUND
26        A + 2
27    U,  A - :M$[- 2], Z
28    N,  JUMP(- 9)
29        A = GLARCO                          " :ARRAYCOUPLE LOCAL ARRAY
30        F = 1                               " ONE ARRAY IN SEGMENT
31        $ = GLTYPE                          " TYPE

32    U,  $ - 1, Z                            " INTEGER?
33    N,  JUMP(3)
34        CHEEN
35        UINVSE2
36        JUMP(12)
37    U,  $ - 2, Z                            " REAL?
38    N,  JUMP(3)
39        CHEEN
40        UINVSE3
41        JUMP(7)
42    U,  $ - 4, Z                            " COMPLEX?
43    N,  JUMP(3)
44        CHEEN
45        UINVSE5
46        JUMP(2)
47        CHEEN                               " BOOLEAN
48        UINVSE4
49        $ = GLARCO                          " :ARRAYCOUPLE LOCAL ARRAY
50        $ = M$                              " :B[0]
51        $ = MS[- 1]                         " DIMENSION
52        $ + :MS[7]                          " SPACE RESERVATION (TWO TIMES
53        SE1                                 " DIMENSION + 7)
54        $ = GLARCO                          " INITIALISATION
55        $ = MS
56        $ = MS[- 1]
```

Array by value.

copies the actual array in a local array.

```
 57          F = 0
 58          MC = G
 59          S - 1, Z
 60      N,  JUMP(- 3)
 61          MC = - G                        " END INDICATOR
 62          G = GLARCO                      " STACK INDICES NEGATIV
 63          G = MG
 64          S = MG[- 1]
 65          S + :MS[2]
 66          G - S
 67          A = MA[0]
 68          S = MA[0], P
 69      Y,  S + MG[0]

 70      Y,  S + MG[1]
 71      Y,  MC = - S
 72      Y,  A + 1
 73      Y,  F + 2
 74      Y,  JUMP(- 7)
 75          S = GLFLOC
 76          A = MS
 77          A = MA
 78          A = MA[- 1]
 79      U,  A - 1, Z                        " ONE DIMENSIONAL ARRAY?
 80      N,  JUMP(4)
 81          G = MC[- 1]
 82          S = MS, Z
 83          SUBC(MS)
 84          JUMP(10)
 85          G = GLTYPE
 86      U,  A - 2, P                        " MORE DIMENSIONAL ARRAY?
 87      N,  F - 8, Z                        " BOOLEAN ARRAY?
 88      Y,  JUMP(4)
 89          G = MC[- 1]
 90          S = MS, P
 91          SUBC(MS)
 92          JUMP(2)
 93          S = MS, Z
 94          SUBCD(:RSE92)
 95          F = 0                           " STACK IMAGINAIRE PART
 96          MC = F
 97          GL23 = - B                      " READ FROM EMPTY SV := FALSE
 98          G = GLFLOC
 99          DO(MG[3])
100          MC = F                          " STACK REAL PART
101          S = GL23, P                     " READ FROM EMPTY SV?
102      Y,  JUMP(57)
103          DO(MD[- 1])
104          G = GLARCO                      " STACK INDICES NEGATIV
105          G = MG
106          S = MG[- 1]
107          S + :MS[2]

108          G - S
109          A = MA[0]
110          S = MA[0], P
111      Y,  S + MG[0]
112      Y,  S + MG[1]
113      Y,  MC = - S
114      Y,  A + 1
```

```
115     Y,  F + 2
116     Y,  JUMP(- 7)
117         S = GLARCO
118         A = MS
119         A = MA[- 1]
120     U,  A - 1, Z
121     N,  JUMP(4)
122         G = MC[- 1]
123         S = S, Z
124         SUBC(MS)
125         JUMP(10)
126         G = GLTYPE
127     U,  A - 2, P
128     N,  F - 8, Z
129     Y,  JUMP(4)
130         G = MC[- 1]
131         S = S, P
132         SUBC(MS)
133         JUMP(2)
134         S = S, Z
135         SUBCD(:PSE92)
136         G = GLTYPE
137         JUMP(G)
138         0
139         JUMP(7)                          " INTEGER
140         JUMP(8)                          " REAL
141         0
142         JUMP(8)                          " COMPLEX
143         0
144         0
145         0
146         JUMP(10)                         " BOOLEAN

147         SUBCD(:PSE38)                    " LEFT SUBSCRIPTED INTEGER
148         JUMP(5)
149         SUBCD(:PSE40)                    " LEFT SUBSCRIPTED REAL
150         JUMP(3)
151         SUBCD(:PSE42)                    " LEFT SUBSCRIPTED COMPLEX
152         F = M[B - 6]
153         GIP = F
154         F = M[B - 4]
155         DO(MC[- 1])
156         JUMP(3)
157         SUBCD(:PSE106)                   " LEFT SUBSCRIPTED BOOLEAN
158         F = M[B - 4]
159         SUBCD(:PSE105)                   " ASSIGN TO BOOLEAN
160         B - 4
161         G = GLARCO                       " INCREASE INDICES
162         G = MG
163         S = MG[- 1]
164         S + :MS[2]
165         G - S
166         DO(MD[- 1])
167         A = MA[0]                        " WP
168         S = MA[0], P
169     N,  JUMP(9)
170         S + 1
171     U,  S + MG, Z
172     Y,  S = 0
173         MA[0] = S
```

```
174      Y,  A + 1
175      Y,  F + 2
176      Y,  JUMP(- 9)                          " INCREASE NEXT INDEX
177          DO(MD[- 1])
178          JUMP(- 117)                        " COPY NEXT ELEMENT
179          DO(MD[- 1])
180          B = MA[0]
181          F = 0
182          GIP = F
183          $ = GLRETURN
184          MC = $
185          $ = 25000
186          GBILL + S, F
187          SE115
188          $ = GLARCO
189          F = M$
190          $ = GLFLOC
191          M$ = F
192          CTERUG

193 UOUTARRAY(19)
194          G = MS[0]
195          $ = -MG[-4]
196          $ + 511
197          $ '*' -511
198          MC = $
199          G = MG[-2]
200          $ = MG[-1]
201          MC = $
202          RU$(9)
203          $ +:MG[-1]
204          SUBCD(!UKILLSEGR)
205          $ = MC[-1]
206          $ - 512,Z
207      N,  JUMP(-7)
208          $ = MC[-1]
209          $ - MG[-1],Z
210      N,  JUMP(-13)
211          DO(MD[-1])
212          CTERUG

213 UINVM23C(5)
214          '035 016 012'
215          '033 135 030'
216          '017 017 135'
217          '031 036 135'
218          '014 174 776'
219 UINVM8C(4)
220          '033 016 135'
221          '014 135 017'
222          '033 016 016'
223          '130 776 000'

224 UINVSE117(22)
225          A = G
226          G = M$
227          A - 1
228          MG[ -3] = - A
229          MG[ -4] = A
230          A + 1
```

```
231         SUBCD(:P91E3[151])          " MERGE WITH INDEXING ROUTINE
232         A '*' - 511
233         A + 512                     " INV. ADDRESS FIRST WORD IN NEXT SV IF PRESENT.
234         A - MG[- 3]
235         MG[- 1] + A                 " FILL NEW CRITERIUM
236         PLUSA(MG[- 3])              " FILL NEW CRITERIUM + POSITION
237         S = 0
238         MG[- 2] = - S               " SET INDICATION LAST LIST
239         LCSA(18)                    " ADDRESS NEXT SV IN S
240         DQ(MD[- 1])                 " RESTORE LRR
241         MA[0] = S                   " RESET WR
242         MA[- 2] = G                 " RESET SV LINK
243         S = MC[- 1]                 " TAKE INVARIANT RETURNADDRESS
244         GL0 = S
245         GL1 = A
246         SUBCD(:RSE34)               " GENERAL GOTO

247 CINVM21(7)
248         '027 030 135'               " NO FRESH REEL ASKED.
249         '017 033 016'
250         '034 021 135'
251         '033 016 016'
252         '025 135 012'
253         '034 024 016'
254         '015 130 776'
255 CINVRLM21(4)                              " NOT READY.
256         '027 030 035'
257         '135 033 016'
258         '012 015 042'
259         '130 776 000'
260 UINVM20A(6)                         " NEW REEL ON RU A.
261         '027 016 040'
262         '135 033 016'
263         '016 025 135'
264         '030 027 135'
265         '031 036 135'
266         '012 130 776'
267 UINVM20B(6)                         " NEW REEL ON RU B.
268         '027 016 040'
269         '135 033 016'
270         '016 025 135'
271         '030 027 135'
272         '031 036 135'
273         '013 130 776'
274 UINVM20C(6)                         " NEW REEL ON RU C.
275         '027 016 040'
276         '135 033 016'
277         '016 025 135'
278         '030 027 135'
279         '031 036 135'
280         '014 130 776'
281 UINVRLM20A(5)                             " INSRECT RLO A.
282         '022 027 034'
283         '031 016 014'
284         '035 135 031'
285         '025 030 135'
286         '012 130 776'
287 UINVM23A(5)                         " TEAR OFF RU A+
288         '035 016 012'
289         '033 135 030'
```

strings of messages

(see comment)

```
290          '017 017 135'
291          '031 036 135'
292          '012 174 776'
293 VINVM23B(5)                          " TEAR OFF RU B*
294          '035 016 012'
295          '033 135 030'
296          '017 017 135'
297          '031 036 135'
298          '013 174 776'

299 VINVTRM26A(7)                        " RU A DISCONNECTED.
300          '031 036 135'
301          '012 135 015'
302          '022 034 014'
303          '030 027 027'
304          '016 014 035'
305          '016 015 130'
306          '776 000 000'
307 VINVTRM26B(7)                        " RU B DISCONNECTED.
308          '031 036 135'
309          '013 135 015'
310          '022 034 014'
311          '030 027 027'
312          '016 014 035'
313          '016 015 130'
314          '776 000 000'
315 VINVTRM26C(7)                        " RU C DISCONNECTED.
316          '031 036 135'
317          '014 135 015'
318          '022 034 014'
319          '030 027 027'
320          '016 014 035'
321          '016 015 130'
322          '776 000 000'
323 VINVPLM26A(7)                          " PLO A DISCONNECTED.
324          '031 025 030'
325          '135 012 135'
326          '015 022 034'
327          '014 030 027'
328          '027 016 014'
329          '035 016 015'
330          '130 776 000'
331 VINVTRM28A(12)                       " RU A OUT OF ORDER, SERIAL NUMBER
332          '031 036 135'
333          '012 135 030'
334          '036 035 135'
335          '030 017 135'
336          '030 033 015'
337          '016 033 127'
338          '135 034 016'
339          '033 022 012'
340          '025 135 027'
341          '036 026 013'
342          '016 033 135'
343          '776 000 000'

344 VINVTRM28B(12)                       " RU B OUT OF ORDER, SERIAL NUMBER
345          '031 036 135'
346          '013 135 030'
347          '036 035 135'
```

```
348           '030 017 135'
349           '030 033 015'
350           '016 033 127'
351           '135 034 016'
352           '033 022 012'
353           '025 135 027'
354           '036 026 013'
355           '016 033 135'
356           '776 000 000'
357 UINVTRM28C(12)                    " RU C OUT OF ORDER, SERIAL NUMBER
358           '031 036 135'
359           '014 135 030'
360           '036 035 135'
361           '030 017 135'
362           '030 033 015'
363           '016 033 127'
364           '135 034 016'
365           '033 022 012'
366           '025 135 027'
367           '036 026 013'
368           '016 033 135'
369           '776 000 000'
370 UINVRLM28A(12)                        " RLO A OUT OF ORDER, SERIAL NUMBER
371           '031 025 030'
372           '135 012 135'
373           '030 036 035'
374           '135 030 017'
375           '135 030 033'
376           '015 016 033'
377           '127 135 034'
378           '016 033 022'
379           '012 025 135'
380           '027 036 026'
381           '013 016 033'
382           '135 776 000'
383 UINVRLM23A(11)                        " RLO A REN JAMMED, SERIAL NUMBER
384           '031 025 030'
385           '135 012 135'
386           '031 016 027'
387           '135 023 012'
388           '026 026 016'
389           '015 127 135'
390           '034 016 033'
391           '022 012 025'
392           '135 027 036'
393           '026 013 016'
394           '033 135 776'
395 UINVBLANK(1)                      " 2 OCTADES BLANK TAPE
396           '000 000 776'
397 UINVTRA(5)                        " A MET 8 OCTADES BLANK TAPE
398           '077 110 110'             followed
399           '110 077 000'               by
400           '000 000 000'
401           '000 000 000'
402           '000 776 000'
403 UINVTRB(5)                        " B MET 8 OCTADES BLANK TAPE
404           '177 111 111'
405           '111 167 000'
406           '000 000 000'
407           '000 000 000'
```

```
408           '000 776 000'
409 UINVTRC(5)                                  " C MET 8 OCTADES BLANK TAPE
410           '034 042 101'
411           '101 101 000'
412           '000 000 000'
413           '000 000 000'
414           '000 776 000'
415 UINVALLHOLE(1)                      " 1 ALL HOLE
416           '177 776 000'
417 UINVALLHOLE4(2)                             " 4 ALL HOLES
418           '177 177 177'
419           '177 776 000'
420 UINVARROW1(3)                               " EIND PIJL
421           '177 177 177'
422           '177 076 034'
423           '010 776 000'
424 UINVARROW2(2)                               " BEGIN PIJL
425           '177 076 034'
426           '010 776 000'

427 CTARERROC1(145)
428           CTARERROC1                        " INVARIANT ADDRESS OF THIS WORD
429           A = - S                           " SNUFFEL(8)
430       U,  A + 10, E                         " DIGIT?
431       N,  A + 64, Z                         " +
432       N,  A + 1, Z                          " -
433       N,  A + 23, Z                         " .
434       N,  A + 1, Z                          " 10
435           A = S
436           GOTO(MC[- 1])
437           B + 6                             " CREATE WORKING SPACE FOR NUM
438           SUBC(:MT[124])
439           SUBC(:MT[- 111])
440       Y,  JUMP(6)
441       U,  A - 120, Z                        " APOS?
442       N,  JUMP(- 5)
443           SUBC(:MT[119])
444       U,  A - 120, Z
445       N,  JUMP(- 3)
446           JUMP(- 9)
447       U,  A - 65, Z                         " MINUS?
448       Y,  M[B - 6] = S                      " NEGMANT := IBUE
449       N,  M[B - 6] = - S
450       N,  A - 64, Z                         " PLUS?
451       Y,  SUBC(:MT[111])
452       Y,  A - 93, Z
453       Y,  JUMP(- 3)
454           SUBC(:MT[- 26])
455       Y,  JUMP(2)
456           S = 2
457           SUBCD(:CALARM)
458           E = 0
459           M[B - 5] = E                      " DEXP := 0; MULT := IBUE
460       U,  A - 89, Z                         " TIENTJE
461           E = 1
462           M[B - 3] = - G                    " POINT := EALSE(= - 1)
463       N,  E = 0
464           M[B - 2] = E                      " BEWAAR MANTISSE
465       Y,  JUMP(36)                          " NA TIENTJE
```

tapereading procedures

like INSYMBOL etc. etc.

```
466     U,  $ - 88, Z                       " PUNT?
467     N,  JUMP(4)
468         A = M[B - 3], P                 " POINT = IRUE?
469     Y,  JUMP(- 14)

470         M[B - 3] = - A
471         SUBC(:MT[91])
472     U,  $ - 9, P
473     Y,  JUMP(- 18)
474         A = M[B - 4], P                 " MULT = IRUE?
475     N,  JUMP(17)                        " TO UPDATE EXPONENT
476         E = M[B - 2]                    " RETRIEVE PREVIOUS E
477         INT                             " NINT := 0
478         E * 10
479         G + S
480         INT
481     Y,  JUMP(9)                         " TO STORE NEW E
482         M[B - 4] = - A                  " MULT := EALSE
483     U,  $ - 4, P                        " TAKE PREVIOUS E
484     N,  JUMP(7)
485         E = M[B - 2]
486         E + 1
487         INT
488     N,  E = CNUMTAB[20]                 " 109 951 162 778
489     N,  A = 1
490     N,  M[B - 5] + A                    " DEXP := DEXP + 1
491         M[B - 2] = E                    " STORE NEW E
492     U,  A = M[B - 4], P                 " MULT
493         A = M[B - 3], E                 " = POINT?
494     Y,  M[B - 5] - A                    " UPDATE DEXP
495         SUBC(:MT[67])
496     U,  $ - 9, P                        " - DIGIT ?
497     N,  JUMP( - 24)
498     U,  $ - 88, Z                       " . ?
499     Y,  JUMP( - 32)
500     U,  $ - 89, Z                       " 10
501     N,  JUMP(29)

502         SUBC(:MT[60])                   " NA TIENTJE
503     U,  $ - 93, Z                       " BLANKS ALLOWED
504     Y,  JUMP(- 3)
505     U,  $ - 65, Z                       " MINUS?
506     Y,  M[B - 4] = S                    " NEGEXP := IRUE
507     N,  M[B - 4] = - S                  " NEGEXP := EALSE
508     N,  $ - 64, Z                       " PLUS ?
509     Y,  SUBC(:MT[53])
510     Y,  S - 93, Z
511     Y,  JUMP(- 3)
512     U,  A - 9, P
513     Y,  JUMP(- 58)
514         E = 0
515         M[B - 3] = G                    " EXPO := 0
516         S = M[B - 3]
517     U,  $ - 32767, P
518     N,  TENS
519     N,  S + A
520     N,  M[B - 3] = S                    " STORE EXPO
521         SUBC(:MT[41])
522     U,  S - 9, P
523     N,  JUMP(- 8)
```

```
524     U,  A - 88, Z                       " PUNT?
525     N,  A - 89, Z                       " TIENTJE?
526     Y,  JUMP(- 71)                      " NIET TOEGESTANE AFSLUITER
527         A = M[B - 4], P
528         A = M[B - 3]
529     Y,  M[B - 5] - A                    " ADD TWO EXPO CONSTITUENTS
530     N,  M[B - 5] + A
531         G = UBLOCKSTR                   " AFSLUITER
532         MG[14] = S                      " PRESYM
533         E = M[B - 2], Z

534     Y,  GOTOR(MC[- 7])
535         S = M[B - 5], Z
536     Y,  JUMP(6)
537         E * 10
538         INT
539     N,  JUMP(6)
540         M[B - 2] = E
541         S - 1, Z
542     N,  JUMP(- 6)
543     U,  A = M[B - 6], P *
544     Y,  E = - E
545         GOTOR(MC[- 7])
546         M[B - 4] = - S, P               " NEGEXP := (DEXP < 0)
547     Y,  S = - S
548     U,  S - 630, P
549     Y,  S = 768
550         E = 1
551         A = :CNUMTAB
552         RCS(1), P
553     N,  E * MA
554         S '*' UCON3. Z                  " 2 ↑ 26 - 1
555     N,  A + 2
556     N,  JUMP(- 5)
557     U,  S = M[B - 4], P
558     N,  E * M[B - 2]
559     Y,  M[B - 4] = E
560     Y,  E = M[B - 2]
561     Y,  E/M[B - 4]
562         JUMP(- 20)

563         SUBC(:MT[71])                   " NSYM(10)
564         A = S, P
565     N,  S = 3
566     N,  SUBCD(:CALARM)
567     U,  S - 93, Z                       " TO SKIP ONE SPACE
568     N,  GOTOR(MC[- 1])
569         SUBC(:MT[65])
570         A = S, P
571     Y,  GOTOR(MC[- 1])
572         JUMP(- 8)

573 COCT2(2)
574         SUBC(:MT[198])
575         JUMP(1)
576 COCT1(10)
577         SUBC(:MT[198])
578         SUBCD(:COCT)
579         A = MG[10]
580     U,  A - 1, Z
```

```
581      N,  MG[10] = A                          " BASINT := 0
582          A = 0
583          MG[14] = - A                        " PRESYM := EMPTY
584          G = S                               " RESULT IN G
585          S = COCTRRICE
586          JUMP(340)                           " EXIT
587  COCTI(2)
588          SUBC(:MT[203])
589          JUMP(- 11)

590  CHARE2(2)
591          SUBC(:MT[184])                      " CKOR2
592          JUMP(1)
593  CHARE1(4)
594          SUBC(:MT[184])                      " CKOR1
595          SUBC(:MT[238])                      " CHARE
596          G = UBLOCKSTR
597          JUMP(336)
598  CHARE!(2)
599          SUBC(:MT[195])                      " CKOR!
600          JUMP(- 5)

601  CNUM!NSR(28)
602          SUBC(:MT[193])
603          B + 1
604          M[B - 1] = - B                      " APOS := EALSE
605          SUBC(:MT[152])                      " SYM!NSPECTION
606          S - 3, Z
607      N,  S + 3, Z
608      N,  JUMP(17)
609          SUBC(:MT[32])
610      U,  S - 120, Z                          " IS HET EEN APOSTROPHE
611      Y,  S = M[B - 1]
612      Y,  M[B - 1] = - S                      " APOS := - APOS
613      Y,  JUMP(- 8)
614      U,  S = M[B - 1], F
615      Y,  JUMP( - 11)
616          SUBC(:MT[  - 181])
617      Y,  A = UBLOCKSTR
618      Y,  MA[14] = S                          " BACKSYM
619      Y,  F = 0
620      Y,  S = CNUM!NSPRICE
621      Y,  JUMP(310)
622          S - 122, Z                          " WAS HET EEN MARK(?)
623      N,  JUMP(- 19)
624          F = 1
625          JUMP(- 6)
626      U,  S - 2, Z                            " KOMT ER EINDE BAND?
627      Y,  F = 2
628      Y,  JUMP(- 9)
629          JUMP(- 25)                          " DUS IS DE CODE AFGEVALLEN

630  CNUM2(2)
631          SUBC(:MT[148])
632          JUMP(1)
633  CNUM1(8)
634          SUBC(:MT[148])
635          SUBC(:MT[- 190])
636          S = UBLOCKSTR
637          A = 1
```

```
638          MS[21] + A                    " AANTAL GESLAAGDE NUMS
639          MS[22] = F                    " WAARDE VAN LAATSTGELEZEN NUM
640          S = CNUMPRICE
641          JUMP(292)

642 CNUM1(113)
643          SUBC(:MT[155])
644          JUMP(- 9)

645          G = UBLOCKSTR                 " CSYM(86)
646          S = MG[14], Z
647     Y,   A = - S, P
648     N,   MG[14] - S
649     N,   GOTOR(MC[- 1])
650          SUBC(:MT[80])
651     U,   S - 122, P
652     N,   GOTOR(MC[- 1])                " SYM AND CHAR SAME REPR. OR SYM := - 1
653     U,   S - 166, P
654     U,   S - 137, E                    " S < 138 ∨ S > 166
655     N,   JUMP(14)
656          SUBC(:MT[88])                 " IETS ALS ≤?
657     Y,   S = A
658     Y,   GOTOR(MC[- 1])
659          SUBC(:MT[71])                 " GET NEXT CHARACTER, WHILE SKIPPING
660          SUBC(:MT[84])
661     Y,   JUMP(4)
662     U,   S - 122, P
663     Y,   JUMP(- 5)
664     U,   S + 1, Z
665     Y,   GOTOR(MC[- 1])
666          G = UBLOCKSTR
667          MG[13] = S                    " BACKCHARE
668          S = - 2
669          GOTOR(MC[- 1])                " DELIVER " ONBEKENDE WORDDELIMITER"
670          A = :MS[- 137]
671          B + 4                         " MAKE ROOM FOR LOCALS
672          S = 0
673          M[B - 2] = S                  " WAARDE := 0
674          F = :CSYMTAB2[8]
675          M[B - 3] = A                  " SAVE WOORD
676          M[B - 1] = G                  " SAVE LISTPOINTER

677          SUBC(:MT[161])                " ORDINARY CALL OF CHARF
678          SUBC(:CORSYM)                 " CORRECTION TO DISALLOW END OF TAPE/CODE IN WORDDELIMITER
679     U,   S - 166, P
680          S - 137, E                    " ¬ ALLOWED UNDERLINED LETTER?
681          A = M[B - 3]
682     N,   JUMP(11)
683          S + 44, Z                     " SPACE?
684     N,   S - 128, Z                    " UNDERLINED SPACE?
685     N,   S = M[B - 4]                  " RESTORE CHARACTER IN S
686     N,   B - 4
687     N,   JUMP(- 28)
688     U,   A - 239, Z                    " WOORD = 239, I.E. GO READ UP TO HERE?
689     Y,   S = M[B - 2], Z               " IN FIRST HALF OF WORDDELIMITER?
690     N,   S = M[B - 4]                  " RESTORE CHARACTER IN S
691     N,   B - 4
692     N,   JUMP(- 27)
693          S = 30                        " REPRESENTATION FOR SPACE IN GOIO
694          LUA(5)                        " SHIFT WORD
```

read dec. number from tape

```
695          A = S                          " ADD NEXT CHARACTER
696          G = M[B - 1]                   " RESTORE LIST POINTER
697          S = MG, R
698     N,   S = - S                        " INVERT COMPARE-WORD
699     U,   A - S, Z                       " THIS IS IT?
700     Y,   JUMP(7)
701          S = MG, R                      " MORE IN THIS PART OF THE LIST
702          F + 1
703     Y,   JUMP(- 7)
704          S + DPT[2], R                  " - END OF THE LIST
705     Y,   JUMP(- 31)                     " M[B - 3] = A
706          B - 4
707          JUMP(- 49)                     " SKIP TILL END OF UNDERLINING

708          F - :CSYMTAB2, Z               " MAAK RANGNUMMER
709     Y,   S = 0                          " GRAAG + 0
710     N,   S = G
711          A = 0                          " SIGN CONSISTENT 0
712          DIVAS(3), Z
713          S + :CSYMTAB3
714          S = MS
715     N,   RUS(9)
716     N,   A - 1, Z
717     N,   RUS(9)
718          S '*' 511
719          A = M[B - 2], Z                " IS DIT NIET EEN STAART-WOORD?
720     N,   JUMP(5)
721     U,   S - 256, R                     " IS HET DAN EEN KOP-WOORD?
722     N,   JUMP(6)
723          M[B - 2] = S                   " BERG TUSSENWAARDE OP
724          F = :CSYMTAB2
725          JUMP(- 51)
726          S - 356
727     U,   S - :MA[- 256], Z              " KLOPPEN KOP EN STAART?
728     N,   JUMP(- 23)
729          B - 4
730          GOTOR(MC[- 1])                 " WORD DELIMITER GEVONDEN

731          SUBC(:MT[107])                 " CSCHARF(14)
732     U,   S + 0, R                       " NIETS AAN DE HAND?
733     Y,   GOTOR(MC[- 1])
734     U,   S + 1, Z                       " EINDE-CODE?
735     N,   S = - 1                        " DAN WAS HET FOUTE PARITEIT O.I.D.
736     N,   GOTOR(MC[- 1])
737          SUBC(:CODEEIN)
738          JUMP(0)
739     Y,   JUMP(- 9)
740     U,   S + 1, Z                       " WAS HET EINDE BAND
741     N,   JUMP(- 7)
742          S = UBLOCKSTR, Z               " DAN BANDAANVRAGE
743          SUBCD(:UNONSTAN)
744          JUMP(- 8)

745          F = :CSYMTAB1                  " CENKEL(11)
746          A = MG, R
747     N,   GOTO(MC[- 1])                  " NIET GEVONDEN, COND, N
748          A '*' 511
749     U,   S - A, Z                       " IS 'T M?
750     N,   F + 1
751     N,   JUMP(- 6)
```

```
752          A = MG
753          RUA(9)
754          A '*' 511
755          GOTO(MC[- 1])                    " GEVONDEN, SYM-WAARDE IN A, COND, Y

756 CSYMINSPRI(69)
757          SUBC(:MT[42])
758          SUBC(:MT[4])                     " KERN VAN DE ROUTINE
759          G = S
760          S = CSYMINSPRICE
761          JUMP(174)
762          G = UBLOCKSTR                    " CSYMINSP(22)
763          S = MG[14], Z                    " PRESYM EMPTY?
764     Y,   A = - S, P
765     N,   JUMP(7)                          " TO DELIVER 0
766          S = MG[10]
767     U,   S - 2, Z                         " BASINT = 2 ?
768     N,   JUMP(6)
769          SUBC(:MT[70])                    " CALL CHARF
770     U,   S + 1, Z
771     Y,   S = 1                            " ER KOMT EEN CODE AFSLUITER
772     N,   MG[13] = S                       " BACKCHARF
773     N,   S = 0
774          GOTOR(MC[ - 1])
775          SUBC(:CODEFIN)
776     Y,   S = 0
777     Y,   GOTOR(MC[ - 1])
778     U,   S + 1, Z                         " END OF TAPE ?
779     Y,   S = 2
780     N,   MG[12] = S                       " BACKOCT
781     N,   S = 3
782          GOTOR(MC[ - 1])

783          S = 2                            " CKOP2(2)
784          JUMP(1)
785          S = 1                            " CKOP1(15)
786          S + UTRSTREAMS
787          S = MS[- 1]
788          UBLOCKSTR = S
789          S = MC[- 1]                      " UNSTACK LINK
790          CGL1 = S                         " AND STORE IN GLOBAL
791          A = 1
792          S = :MA[3]
793          SE18                             " CREATE DISPLAY
794          S = MA[- 6]
795     U,   S - 9, Z                         " NO PARAMETERS?
796     N,   SUBCD(:RSE19)                    " ALARM WRONG NO, OF PASAN
797          S = 16
798          SE1                              " ANONYMONS STACK SPACE
799          GOTOR(CGL1)

800          S = MC[- 1]                      " CKOP1(26), UNSTACK LINK
801          CGL1 = S                         " SAVE LINK IN GLOBAL IN STACK BOTTOM
802          A = 1
803          S = :MA[3]
804          SE18                             " CREATE DISPLAY
805          S = MA[- 6]
806     U,   S - 13, Z
807     N,   SUBCD(:RSE19)                    " ALARM WRONG NR, OF PARAMETERS
808          E = :MA[- 11]
```

```
809           $UBCD(:PSE20)                    " CHECK FORMAL ARITHMETIC
810           S = 16
811           SE1                              " ANONYMONS WORKING SPACE
812           S = CGL1
813           MC = S
814           DOS(MA[- 11])                    " AVALUATE PARAMETER
815           DO(ERV)                          " PARAMETER IN E
816           $UBCD(:PSE88)                    " ROUND PARAMETER
817           S = G
818      U,   S - UNUMTRPMSTR, P
819      N,   S - 0, E
820      Y,   S = 1
821      Y,   $UBCD(:CALARM)                   " NUMBER OF EXISTENT STREAM
822           S + UTRSTREAMS
823           G = MS[- 1]
824           UBLOCKSTR = G
825           JUMP(119)

826 CSYM2(2)
827           SUBC(:MT[- 44])
828           JUMP(1)
829 CSYM1(5)
830           SUBC(:MT[- 44])
831           SUBC(:MT[- 184])
832           G = S
833           S = CSYMPRICE
834           JUMP(103)

835 SYM(116)
836           S = UTRSTREAMS                   " AS CALLED BY TRANSLATOR
837           G = MS[0]
838           UBLOCKSTR = G
839           SUBC(:MT[- 190])                 " CSYM[1]
840           DO(MD[ - 1])
841           CTERUG
842           0                                " SKIP

843           G = UBLOCKSTR                    " CHARF(96)
844           S = MG[10]
845      U,   S - 2, Z
846      N,   S = - 1
847      N,   GOTOR(MC[- 1])
848           S = MG[13], Z
849           MG[13] - S                       " VOOR CHARF := EMPTY
850      Y,   A = - S, P
851      N,   JUMP(83)                         " TO DELIVER VOOR CHARF
852           MC = - S                         " STACK "STRUNDER" AS + 0
853      U,   S = MG[20], P                    " SPATEL IS - 0 PREFERENT
854      Y,   JUMP(41)
855           $UBCD(:COCT)                     " GET OCTAL FROM TAPE
856           S - 127, Z                       " ERASE?
857      N,   S + 127, Z                       " IF NOT, RESTORE; BLANK?
858      Y,   JUMP(- 4)                        " IF SO, SKIP
859      U,   S - 122, Z                       " LOWER CASE?
860      Y,   MG[15] = S                       " LOWER := IRUE
861      Y,   JUMP(- 7)
862      U,   S - 124, Z                       " UPPER CASE?
863      Y,   MG[15] = - S                     " LOWER := EALSE
864      Y,   JUMP(- 10)
865      U,   S - 26, Z                        " NLCR?
```

basic
read symbol from tape

```
866          A = 0
867      Y,  MG[16] = A                    " CARRIAGE POSITION := 0
868      Y,  JUMP(4)
869      U,  $ - 11, Z                     " STOPCODE?
870      Y,  JUMP(2)
871      U,  $ + 1, Z                      " EOT?
872      N,  JUMP(8)
873          MG[20] = - A                  " KAN WEG, WORDT BIJ INITIALISATIE VOOR CHARE := - 0
874          A = M[B - 1], Z               " GEEN STRUNDER HANGEND?

875      N,  MG[12] = $                    " BACKOCT
876      N,  JUMP(21)                      " TO DELIVER A SPACE
877      U,  $ - 11, P                     " NLCR?
878      Y,  $ = 119                       " MAKE CHAR FOR NLCR
879      N,  $ = - 1                       " FOR STOPCODE AND EOT
880          JUMP(53)
881      U,  $ - 16, Z                     " SPACE?
882      Y,  $ = 1
883      Y,  JUMP(8)
884      U,  $ - 62, Z                     " TAB?
885      N,  JUMP(14)
886          $ = MG[16]                    " CARRIAGEPOSITION
887          $ '*' 7                       " REM(POS8)
888      U,  $ - 7, Z                      " THEN GIVE 9 SPACES
889      Y,  $ = 9
890      N,  $ = - $
891      N,  $ + 8                         " ELSE(8 - REM(POS8)) SPACES
892          MG[16] + $
893          MG[20] + $
894      U,  A = M[B - 1], Z               " GEEN NOTITIE?
895      Y,  JUMP(- 41)                    " TO GET NEXT HEPTAD
896          $ = 1
897          MG[20] - $
898          $ = 93, Z                     " INTERNAL REPR. OF SPACE
899          JUMP(31)
900      U,  A = MG[20], P
901      Y,  MG[12] = $
902      Y,  JUMP(- 7)
903      U,  $ - 14, Z                     " UNDER OF STROKE
904      N,  JUMP(6)
905      U,  A = MG[15], P
906      Y,  $ = - 128                     " MAKE UNDERLINING
907      N,  $ = - 256                     " MAKE STROKE

908          $ '*' - M[B - 1]
909          M[B - 1] = - $
910          JUMP(- 56)                    " TO GET NEXT HEPTAD
911          A = 1
912          MG[16] + A
913      U,  $ - 127, P                    " NON EXISTENT FLEXOCODE?
914      Y,  JUMP(18)
915          A = MG[15], P
916      N,  $ '+' 127                     " IE NON LOWERCASE THEN 127 + $
917          A = 0
918          DIVAS(3), Z
919          $ + :CHARETAB
920          $ = MS                        " GET WORD FROM TABLE
921      N,  RUS(9)
922      N,  A - 1, Z
923      N,  RUS(9)
```

```
924           $ '*' 511                          " COLLATE VALUE
925           A = MG[15], R
926       Y,  JUMP(3)
927       U,  $ - 127, R
928       N,  $ = - 2
929       Y,  $ '*' 127                          " SUBTRACT 128, LEAVING + 0
930       U,  $ - 126, R                         " WRONG PARITY(>127) OR NON EXISTENT(=127)
931       N,  $ + MC[- 1]
932       N,  GOTOR(MC[- 1])
933           $ = - 2
934           B - 1
935       U,  $ + 1, Z
936       Y,  A = 0
937       Y,  MG[10] = - A
938           GOTOR(MC[- 1])

939           GBILL + S, R     " MAKE ACCOUNT
940           SE115
941           DO(MD[- 1])                        " RESTORE LRP
942           SE 71                              " PROCEDURE EXIT

943           A = 0                              " END OF CHARF
944           MG[14] = - A                       " PRESYM := EMPTY
945           G = $                              " RESULT IN G
946           $ = CHARERRICE
947           JUMP(- 9)
948           $ = MC[ - 1]
949           $ '*' 511
950           $ - 510
951           JUMP($)                            " RETURN ON SAME SEGMENT
952 CSYMI(2)
953           SUBC(:MT[ - 150])
954           JUMP( - 122)

955 UINVTIME(6)
956           '035 022 026'                      " TIM
957           '016 101 036'                      " E-U
958           '027 022 035'                      " NIT
959           '034 135 036'                      " S U
960           '034 016 015'                      " SED
961           '130 776 000'                      " .

962 CTARERROC2(1)
963           CTARERROC2

964 CODEDEF(51)
965           SUBC(:MT[1])
966           CTERUG
967           SUBCD(:COCT)
968       U,  $ - 31, Z
969       N,  JUMP(7)
970           SUBCD(:COCT)
971       U,  $ - 31, Z                          " SKIPPING LEADING "VIJFERASE"
972       Y,  JUMP(- 3)
973       U,  $ + 0, Z                           " SKIPPING LEADING BLANK TAPE"
974       Y,  SUBCD(:COCT)
975       Y,  JUMP(- 3)
976           A = 0, Z                           " TO MAKE COND. Y
977           SUBC(:MT[31])                      " TO SKIP POSSIBLE FLEXONOISE
978       N,  A = 0
```

more tape reading procedures

```
 979      U,  $ - 26, Z                    " NLCR?
 980      Y,  MG[15] = $                   " INITIALISE FLEXO-CASE AS LOWER (CONVENTION)
 981      Y,  JUMP(6)
 982      U,  $ - 122, Z                   " FLEXOLOWER?
 983      Y,  JUMP(4)
 984      U,  $ - 124, Z                   " FLEXOUPPER?
 985      N,  A + 0, R
 986      N,  GOTO(MC[- 1])
 987      U,  $ - 124, Z                   " TO DIFFERENTIATE THE CONDITION
 988      Y,  MG[12] = $                   " BACKOCT
 989      Y,  $ = 2
 990      Y,  A = 0
 991      Y,  MG[16] = A
 992      Y,  MG[20] = - A
 993      Y,  MG[13] = - A
 994      Y,  MG[10] = $
 995      Y,  GOTO(MC[- 1])
 996      U,  $ - 31, Z                    " SKIPPING VIJFERASE = LETTERS
 997      Y,  SUBCD(:COCT)
 998      Y,  JUMP(- 3)                    " CARRIAGE RETURN?(TELEX)
 999          MG[15] = $                   " TELEX LETTERS
1000      U,  $ - 2, Z

1001      Y,  JUMP(5)
1002      U,  $ - 8, Z                     " NEW LINE (TELEX)
1003      Y,  JUMP(3)
1004      U,  $ - 27, Z                    " CIJFERS?
1005      N,  GOTO(MC[- 1])
1006          MG[15] = - $
1007          $ = 3                        " BASINT VOOR TELEXCODE
1008          JUMP(- 19)
1009      U,  $ - 11, Z                    " SKIPPING FLEXONOISE
1010      N,  $ - 127, Z
1011      N,  $ + 127, Z
1012      N,  GOTOR(MC[- 1])
1013          SUBCD(:COCT)
1014          A = - 0
1015          JUMP(- 7)

1016 CBACKCHARF(6)
1017          SUBC(:MT[117])
1018          G = UBLOCKSTR
1019          A = MG[10]
1020      U,  A - 2, Z
1021      Y,  MG[13] = $
1022          JUMP(12)

1023 CBACKSYM(4)
1024          SUBC(:MT[111])
1025          G = UBLOCKSTR
1026          MG[14] = $
1027          JUMP(8)

1028 CBACKOCT(13)
1029          SUBC(:MT[107])
1030          G = UBLOCKSTR
1031      U,  A = MG[7], R                 " STREAM STILL ALIVE
1032      N,  JUMP(4)
1033      U,  $ - 255, R
1034      U,  $ - 0, E                     " IMPOSSIBLE VALUE OF OCT
```

```
1035      Y,   $ = 0
1036           MG[12] = $
1037           $ = 4                          " STANDARD PRICE OF SIMILAR PROCEDURES
1038           GBILL + $, P
1039           $E115
1040           DO(MD[- 1])
1041           $E71

1042 C|NICHARE(48)
1043           A = 1
1044           $ = :MA[3]
1045           $E18                           " CREATE DISPLAY
1046           $ = MA[- 6]
1047      U,   $ - 21, Z                      " 3 PARAMETERS?
1048      N,   $UBCD(:BSE19)                  " WRONG NR. OF PARAMETERS
1049           F = :MA[- 19]
1050           $UBCD(:BSE20)                       " CHECK FORMAL ARITHMETIC
1051           F = :MA[- 15]
1052           $UBCD(:BSE20)
1053           F = :MA[- 11]
1054           $UBCD(:BSE21     )                  " CHECK FORMAL BOOLEAN
1055           $ = 12
1056           $E1                            " ANONYMONS RESERVATION
1057           DOS(MA[- 19])                  " EVALUATE !
1058           DO(ERV)                        " ROUND TO INTEGER
1059           $UBCD(:BSE88)
1060           $ = G
1061      U,   $ - UNUMTREAMSTR, P
1062      W,   $ - 0, E
1063      Y,   $ = 1
1064      Y,   $UBCD(:CALARM)
1065           $ + UTRSTREAMS
1066           $ = MS[- 1]
1067           UBLOCKSTR = $
1068           DOS(MA[- 15])
1069           DO(ERV)                        " EVALUATE CARRIAGE POSITION
1070           $UBCD(:BSE88)
1071           $ = G
1072      U,   $ - 144, P
1073      U,   $ - 0, E
1074      Y,   $ = 0
1075           G = UBLOCKSTR
1076           MG[16] = $                     " CARRIAGE POSITION
1077           MG[20] = - $                   " SPATEL
1078           DOS(MA[- 11])
1079           DO(ERV)

1080           $ = UBLOCKSTR
1081           MS[15] = G
1082           F = 0
1083           MS[13] = - G
1084           MS[20] = - G
1085           F = 2
1086           MS[10] = G
1087      U,   $ = MS[7], P                   " STREAM STILL ALIVE?
1088      N,   $UBCD(:UNONSTAN)
1089           $ = 5
1090           JUMP(- 52)                     " AFREKENEN

1091 C|NIDATA(4)                             " INITIALISE PAPER TAPE
```

```
1092          SUBC(:MT[20])
1093      U,  $ = MG[7], R                    " STREAM ALIVE?
1094      N,  SUBCD(:UNONSTAN)                " DAN AANVRAGEN
1095          JUMP(- 57)

1096 CSKIPRATA(78)
1097          SUBC(:MT[16])
1098          $ = 0
1099          MG[6] = - $                     " INTEREST := FALSE
1100          MC = $
1101          $ = 2
1102          M[B - 1] + $                    " ACCOUNT
1103          SUBCD(:COCT)
1104          $ + 1, Z
1105      Y,  JUMP(- 5)
1106          MG[ 6] = - $                    " INTEREST := TRUE
1107          MG[14] = $                      " PRESYM
1108          MG[10] = $                      " BASINT
1109          MG[20] = $                      " SPATEL
1110          $ = MC[- 1]                     " UNSTACK ACCOUNT
1111          JUMP(- 71)
1112          0                               " SKIP
1113          0                               " SKIP

1114          $ = MC[- 1]                     " CKOPI(26), UNSTACK LINK
1115          CGL1 = $                        " SAVE LINK IN GLOBAL IN STACK BOTTOM
1116          A = 1
1117          $ = :MA[3]
1118          SE18                            " CREATE DISPLAY
1119          $ = MA[- 6]
1120      U,  $ - 13, Z
1121      N,  SUBC(:PSE19)                    " ALARM WRONG NR. OF PARAMETERS
1122          E = :MA[- 11]
1123          SUBCD(:PSE20)                   " CHECK FORMAL ARITHMETIC
1124          $ = 16
1125          SE1                             " ANONYMONS WORKING SPACE
1126          $ = CGL1
1127          MC = $
1128          DOS(MA[- 11])                   " EVALUATE PARAMETER
1129          DO(ERV)                         " PARAMETER IN E
1130          SUBCD(:PSE88)                   " ROUND PARAMETER
1131          $ = G
1132      U,  $ - UNUMTRRMSTR, R
1133      N,  $ - 0, E
1134      Y,  $ = 1
1135      Y,  SUBCD(:CALARM)                  " NUMBER OF EXISTENT STREAM
1136          $ + UTRSTREAMS
1137          G = MS[- 1]
1138          UBLOCKSTR = G
1139          JUMP(31)

1140          $ = MC[- 1]                     " CKOPI2(32)
1141          CGL1 = $
1142          A = 1
1143          $ = :MA[3]
1144          SE18
1145          $ = MA[- 6]
1146      U,  $ - 17, Z
1147      N,  SUBCD(:PSE19)                   " WRONG NO. OF PARAM.
1148          E = :MA[- 15]
```

```
1149            $UBCD(:RSE20)                   " CHECK FORMAL ARITHMETIC
1150            $ = :MA[- 11]
1151            SUBCD(:RSE20)                   " CHECK FORMAL
1152            $ = 16
1153            SE1
1154            $ = GGL1
1155            MC = $                          " RESTACK LINK
1156            DOS(MA[- 15])
1157            DO(ERV)
1158            SUBCD(:RSE88)
1159            $ = G
1160      U,    $ - UNUMTRRMSTR, R
1161      N,    $ - 0, E
1162      Y,    $ = 1
1163      Y,    SUBCD(:CALARM)
1164            $ + UTRSTREAMS
1165            $ = MS[- 1]
1166            UBLOCKSTR = S
1167            DOS(MA[- 11])
1168            DO(ERV)
1169            SUBCD(:RSE88)                   " ROUND SECOND PARAMETER
1170            $ = G
1171            A = MC[ - 1]
1172            A '*' 511
1173            A - 205
1174            JUMP(A)

1175 UINVSESEG(1)
1176            UINVSESEG

1177 UINVSE2(5)
1178            SUBCD(:MT[18])
1179            + 1
1180            LUA(0)
1181            :R90E31
1182            + 16
1183 UINVSE3(5)
1184            SUBCD(:MT[13])
1185            + 2
1186            LUA(1)
1187            :R90E32
1188            + 16
1189 UINVSE4(4)
1190            SUBCD(:MT[8])
1191            + 1
1192            SUBCD(:R92E1)
1193            - 19
1194 UINVSE5(219)
1195            SUBCD(:MT[4])
1196            + 4
1197            LUA(2)
1198            :R90E33
1199            + 8
1200            GL3 = A                         " SAVE ADDRESS FIRST ARRAY COUPLE
1201            GL4 = G                         " SAVE NUMBER OF ARRAY IDENTIFIERS
1202            GL5 = B                         " SAVE :B[0]
1203            F = M[B - 2]                    " SAVE INVARIANT RETURN ADDRESS IN
1204            GL6 = F                         " GL6 AND TYPE ARRAY IN GL7
1205            G = MG[0]                       " SIZE
1206            M[B - 2] = - F                  " INITIALIZE B[- 2] WITH = 0
```

" ARRAY DECLARATION

```
1207          M[B - 1] = - E                  " INITIALIZE B[- 1] WITH - 0
1208          DO(MD[- 1])                     " RESTORE A
1209          S = MA[0]                       " VR
1210          GL18 = E
1211          A = 1
1212          M[B - 1] + A                    " INCREASE DIMENSION
1213          G = MS[1]                       " UPPERBOUND
1214          E + 1
1215          A = M[57],Z                     "  UB + 1 ≤ 2↑26?
1216     N,   E = 203
1217     N,   SUBCD(:MT[186])
1218          MS[1] = G                       " STORE UB + 1 ( - 0 PREFERENCE)
1219          G - MS[0],P                     " LB ≤ UB ?
1220     N,   E= 200
1221     N,   SUBCD(:MT[182])
1222          MS[0] = -G                      " STORE LB -UB +1
1223          E * GL18
1224          S + 2
1225     U,   S - :MC[- 2], Z
1226     N,   JUMP(- 17)
1227          GL18 = E                        " GL18 := 0, GL19 := NUMBER OF WORDS PER ARRAY UNPACKED
1228          G * GL4                         " * NUMBER OF ARRAY IDENTIFIERS
1229          A = M[57], Z
1230     N,   SUBCD(:MT[172])
1231          GL20 = G                        " NUMBER OF WORDS PER SEGMENT UNPACKED
1232          S = G
1233          G = GL7                         " RESTORE TYPE
1234     U,   S = MG[2], P                    " INTEGER, REAL OR COMPLEX ARRAY?
1235     N,   DIVAS(27), Z
1236     N,   S + 1
1237          GL21 = S                        " NUMBER OF WORDS PER SEGMENT PACKED
1238          S = M[B - 1]                    " DIMENSION
1239     U,   S - 1, Z                        " ONE DIMENSIONAL ARRAY?
1240     Y,   JUMP(17)
1241          A = - MG[2], P                  " BOOLEAN ARRAY?
1242     N,   S - 2, P                        " DIMENSION > 2?
1243     N,   JUMP(47)
1244          S = - M[B - 4]                  " LAST RANGE
1245          GL19 = S                        " INCREMENT ADDED TO NEXT ARRAY COUPLE
1246          S = MT[5]
1247          MC[0] = S                       " FILL B[0]
1248          S = MG[1]
1249          MC[0] = S                       " FILL B[1]
1250          SUBCD(:MT[115])                 " MAKE SV CHAIN
1251          JUMP(97)
1252          GOTO(:ONZIN10)                  " B[0] IN MAPPING TABLE ARRAY
1253     Y,   GOTO(:ONZIN8)                   " B[0] IN MAPPING TABLE VECTOR
1254          G + MT[- 5], P                  " B[1] IN MAPPING TABLE VECTOR
1255          G + MT[- 7], E                  " B[2] IN MAPPING TABLE VECTOR
1256     N,   GOTO(:R91E3[2])                 " B[3] IN MAPPING TABLE VECTOR
1257          GOTO(:ONZIN9)                   " B[4] IN MAPPING TABLE VECTOR
1258          S = MT[- 6]
1259          MC[0] = S                       " FILL B[0]
1260          A = MG[0]                       " SIZE
1261          E = MT[- 8]
1262          MC[0] = E                       " FILL B[1] AND B[2]
1263          E = MT[- 8]
1264          MC[0] = E                       " FILL B[3] AND B[4]
1265          RUA(1)                          " 0, 1, 0, 2 INTEGER, REAL, BOOLEAN, COMPLEX
1266          M[B - 2]- A                     " MODIFY B[3]
```

```
1267          $ = GL20                          " NUMBE OF WORDS PER SEGMENT UNPACKED
1268     U,   $ - UVECTORTJE, R                 " BIG VECTOR?
1269     Y,   JUMP(8)
1270          SUBCD(:SRR)                       " STACK PAGE REQUEST
1271     N,   B = NSP
1272          GL18 = B                          " SPECIFIC CONSTANT FIRST ARRAY COUPLE
1273          G = UCLEAR
1274          MC[0] = G                         " INITIALIZE VECTOR IN STACK
1275          $ - 1, Z
1276     N,   JUMP(-3)
1277          JUMP(71)
1278          SUBCD(:MT[87])                    " MAKE SV CHAIN
1279          $ = GL7                           " RESTORE TYPE
1280          $ = M$[2], R                      " INTEGER, REAL OR COMPLEX ARRAY
1281     Y,   GL18 = A, R                       " SPECIFIC CONSTANT FIRST ARRAY COUPLE
1282     Y,   A = 5
1283     N,   A = - S, R                        " BOOLEAN ARRAY?
1284     N,   A = 10
1285          MG[3] + A                         " MODIFY B[3]
1286          JUMP(62)
1287     Y,   GOTO(:RSE90)                      " B[0] HORIZONTAL MATRIX
1288          GOTO(:ONZIN5)                     " B[1] HORIZONTAL MATRIX
1289     Y,   GOTO(:R90E21)                     " B[0] VERTICAL MATRIX
1290          GOTO(:ONZIN5)                     " B[1] VERTICAL MATRIX
1291          F = MG[1]
1292          MC[2] = F                         " FILL B[2] AND B[3]
1293          $ = - M[B - 8]                    " RANGE FIRST INDEX
1294     U,   $ + M[B - 6], R                   " FIRST RANGE > SECOND RANGE?
1295     Y,   F = MT[- 7]                       " VERTICAL MATRIX
1296     N,   F = MT[- 10]                      " HORIZONTAL MATRIX
1297          MC[- 2] = F                       " FILL B[0] AND B[1]
1298          A = - M[B - 8]                    " RANGE SECOND INDEX
1299     Y,   LGSA(27)                          " S := CHOPPED, A := SHIFTED RANGE
1300          GL19 = A                          " INCREMENT ADDED TO NEXT ARRAY COUPLE
1301          MC[1] = S                         " B[5] := I
1302          G = GL7                           " RESTORE TYPE
1303          A = 0
1304          DIVAS(MG[3])                      " A := WIDTH, S := NUMBER UPPERLAYERS
1305          MC[0] = A                         " B[5] := IC
1306     U,   $ + UOVERSHOOT[- 510], R          " C - TABLE GREATER THEN STACKPAGE?
1307     Y,   SUBCD(:MT[95])
1308          GL7 = S                           " SAVE NUMBER OF UPPERLAYERS
1309     U,   A - 3, R                          " MAKE ENTRY ADDRESS IN
1310     N,   A + 41                            " MULTIPLICATION TABLE
1311     U,   A - 5, R
1312     N,   A + 48
1313     U,   A - 8, R
1314     U,   A - 12, E
1315     N,   A + 51
1316     U,   A - 32, R
1317     Y,   A '+' 63, Z
1318     Y,   A = 18
1319          A + :R90E0[- 1]
1320          M[B - 2] = A                      " FILL B[4]
1321          S = MG[0]                         " SIZE
1322          G = MG[3]                         " WIDTHC
1323          G * S                             " WIDTHC * SIZE
1324          G * GL19                          " WIDTHC * SIZE * SHIFTED RANGE
1325          G * GL4                           " J * WIDTHC * SIZE
1326          MC[1] = G                         " SAVE INCREMENT MULTIPLICATION TABLE IN B[7]
```

```
1327          G * GL7                          " J * IC * SIZE
1328          MUL$(- M[B - 2])                 " - IC * SIZE
1329          S + G                            " (- IC + J * IC) * SIZE
1330          MC[- 1] = S                      " FILL POSITION BOUNDARY LAYER IN B[6]
1331          SUBCD(:MT[34])                   " MAKE SV CHAIN
1332      N,  S = MG[2]
1333      N,  S - MT[31]
1334      N,  MG[2] = - S                      " MODIFY B[2]
1335      N,  S = 7
1336      N,  MG[3] + S                        " MODIFY B[3]
1337      Y,  MG[6] + A                        " MODIFY B[6]
1338          S = GL7, Z                       " NUMBER OF UPPERLAYERS ZERO?
1339      N,  SUBCD(:SPR)                      " STACK PAGE REQUEST
1340      N,  B = NSP                          " CONSTRUCTION C - TABLE
1341          S + B
1342      U,  S - B, Z
1343      N,  S - 1
1344      N,  MS[0] = A
1345      N,  A + MG[7]
1346      N,  JUMP(- 5)
1347          B + GL7
1348          MG[7] = B                        " FILL B[7]
1349          G = GL3                          " ADDRESS FIRST ARRAY COUPLE
1350          A = GL18                         " SPECIFIC CONSTANT FIRST ARRAY COUPLE
1351          S = GL5                          " :B[0]
1352          MG[0] = S                        " FILL FIRST WORD IN ARRAY COUPLE
1353          MG[1] = A                        " FILL SECOND WORD IN ARRAY COUPLE
1354          S = 1
1355          GL4 - S, Z                       " LAST ARRAY IN SEGMENT?
1356      N,  F + 2
1357      N,  A + GL19                         " ADD INCREMENT FOR NEXT ARRAY COUPLE
1358      N,  JUMP(- 8)
1359          DO(MD[- 1])                      " RESTORE A
1360          MA[0] = B                        " SET WR
1361          S = GL6                          " INVARIANT RETURN ADDRESS
1362          MC[0] = S
1363          UNEAR                            " HEARING
1364          CTERUG
1365          :R90E0['4020000074']
1366          S = MC[- 1]                      " STARTING POINT SUBROUTINE MAKE SV CHAIN
1367          GL20 = S                         " SAVE LINK
1368          S = 4                            " LENGTH MINIMUM SV LIST
1369          SUBCD(:SPR)                      " SPACE IN CURRENT STACKPAGE FOR ONE LIST?
1370      N,  B = NSP
1371      N,  S = 507[- UOVERSHOOT]            " MAXIMUM NUMBER OF SV'S IN NEW PAGE
1372      Y,  S = - :MC[UOVERSHOOT + 2]
1373      Y,  S '*' 511                        " MAXIMUM NUMBER F SV'S IN CURRENT PAGE
1374          LUS(9)                           " MAXIMUM NUMBER OF WORDS IN THIS LIST
1375          GL21 - S, P                      " MORE LISTS NEEDED?
1376      N,  S + GL21
1377          S + 511
1378          S '*' -511
1379          DO(MD[- 1])                      " RESTORE A
1380          G = MA[- 2]                      " SV LINK
1381          MC[0] = S                        " FILL P[- 3] WITH CRITERIUM
1382          MC[0] = - F                      " INITIALIZE P[- 2] WITH - 0
1383          M[B - 1] = S                     " FILL P[- 1] WITH CRITERIUM
1384          MA[- 2] = B                      " NEW SV LINK
1385          MG[- 2] = B                      " CHAIN ONTO PREVIOUS LIST
1386          G = GL5                          " :B[0] IN MAPPING TABLE
```

```
1387          A = MG[- 2], Z                    " CHAINING ON FIRST LIST OF SEGMENT?
1388      Y,  MG[- 2] = B                       " FILL B[- 2] IN MAPPING TABLE
1389          A = B
1390          LCA(9)                            " POSITION
1391          M[B - 3] + A                      " MODIFY B[- 3]
1392          A = - 0
1393          MC[0] = A
1394          $ - 512, Z
1395      N,  JUMP(- 3)
1396          $ = GL21, B                       " MORE LISTS NEEDED?
1397      Y,  JUMP(-30)
1398          $ = MG[- 2]                       " :B[0] FIRST LIST
1399      U,  $ = MS[- 2], Z                    " IS THIS THE ONLY LIST?
1400      Y,  A = - MS[- 1]
1401      Y,  A + MS[- 3]                       " POSITION FIRST LIST IN A
1402          GOTO(GL20)                        " RETURN WITH YES CONDITION IF ONE LIST

1403          B = 201
1404          SUBCD(:URROEANELINK)
1405          A = G
1406          $ = GL6
1407          $'*'511
1408          M[B-1] = $
1409          $ = GL6
1410          GL26 = $
1411          RUS(9)
1412          $ '*' DRT[1]
1413          GOTO(:UD200)

1414  CRAPERTORN(7)
1415              '031 033 022'                 " RRI
1416              '027 035 016'                 " NTE
1417              '033 135 031'                 " R R
1418              '012 031 016'                 " ARE
1419              '033 135 035'                 " R T
1420              '030 033 027'                 " ORN
1421              '130 776 000'                 " .
1422  CRAPERLOW(7)
1423              '031 033 022'                 " RRI
1424              '027 035 016'                 " NTE
1425              '033 135 031'                 " R R
1426              '012 031 016'                 " ARE
1427              '033 135 025'                 " R L
1428              '030 040 130'                 " OW.
1429              '776 000 000'
1430  CYOKEOPEN(7)
1431              '031 033 022'                 " RRI
1432              '027 035 016'                 " NTE
1433              '033 135 042'                 " R Y
1434              '030 024 016'                 " OWE
1435              '135 030 031'                 "  OR
1436              '016 027 130'                 " EN.
1437              '776 000 000'
1438  CRARITY(6)
1439              '031 033 022'                 " RRI
1440              '027 035 016'                 " NTE
1441              '033 135 031'                 " R R
1442              '012 033 022'                 " ARI
1443              '035 042 130'                 " TY.
1444              '776 000 000'
```

end of array decl.

messages (line printer)

```
1445 CNBKMES(6)
1446          '031 033 022'                    " PRI
1447          '027 035 016'                    " NTE
1448          '033 135 015'                    " R D
1449          '022 034 030'                    " ISQ
1450          '033 015 016'                    " RDE
1451          '033 130 776'                    " R.

1452 CINVM30(6)
1453          '016 026 031'                    " EMPTY OR PROGEND.
1454          '035 042 135'
1455          '030 033 135'
1456          '031 033 030'
1457          '020 016 027'
1458          '015 130 776'
1459 CINVM31(3)                                " BLOCKED
1460          '013 025 030'
1461          '014 024 016'
1462          '015 130 776'
1463 CINVM32(5)
1464          '027 030 035'                    " NOT BLOCKED.
1465          '135 013 025'
1466          '030 014 024'
1467          '016 015 130'
1468          '776 000 000'
1469 CINVMPOOL(2)                              " POOL
1470          '031 030 030'
1471          '025 135 776'
1472 CINVMINPUT(3)
1473          '022 027 031'                    " INPUT
1474          '036 035 135'
1475          '776 000 000'
1476 CINVMFREE(3)
1477          '017 033 016'                    " FREE
1478          '016 135 135'
1479          '776 000 000'
1480 UINVM361(2)
1481          '031 026 135'
1482          '001 135 776'
1483 UINVM362(2)
1484          '031 026 135'
1485          '002 135 776'
1486 UINVM363(2)
1487          '031 026 135'
1488          '003 135 776'
1489 UINVM364(2)
1490          '031 026 135'
1491          '004 135 776'
1492 UINVM365(2)
1493          '031 026 135'
1494          '005 135 776'
1495 UINVM351(6)
1496          '031 026 135'                    " PM1 STACK USED
1497          '001 135 034'
1498          '035 012 014'
1499          '024 135 036'
1500          '034 016 015'
1501          '135 776 000'
1502 UINVM352(6)
1503          '031 026 135'                    " PM2 STACK USED
```

```
1504          '002 135 034'
1505          '035 012 014'
1506          '024 135 036'
1507          '034 016 015'
1508          '135 776 000'
1509 UINVM353(6)
1510          '031 026 135'                    " RM3 STACK USED
1511          '003 135 034'
1512          '035 012 014'
1513          '024 135 036'
1514          '034 016 015'
1515          '135 776 000'
1516 UINVM354(6)
1517          '031 026 135'                    " RM4 STACK USED
1518          '004 135 034'
1519          '035 012 014'
1520          '024 135 036'
1521          '034 016 015'
1522          '135 776 000'
1523 UINVM355(6)                                    "RM5 STACK USED
1524          '031 026 135'
1525          '005 135 034'
1526          '035 012 014'
1527          '024 135 036'
1528          '034 016 015'
1529          '135 776 000'
1530 UINVM341(7)
1531          '031 026 135'                    " RM1 STACK GRANTED
1532          '001 135 034'
1533          '035 012 014'
1534          '024 135 020'
1535          '033 012 027'
1536          '035 016 015'
1537          '135 776 000'
1538 UINVM342(7)
1539          '031 026 135'                    " RM2 STACK GRANTED
1540          '002 135 034'
1541          '035 012 014'
1542          '024 135 020'
1543          '033 012 027'
1544          '035 016 015'
1545          '135 776 000'
1546 UINVM343(7)
1547          '031 026 135'                    " RM3 STACK GRANTED
1548          '003 135 034'
1549          '035 012 014'
1550          '024 135 020'
1551          '033 012 027'
1552          '035 016 015'
1553          '135 776 000'
1554 UINVM344(7)
1555          '031 026 135'                    " RM4 STACK GRANTED
1556          '004 135 034'
1557          '035 012 014'
1558          '024 135 020'
1559          '033 012 027'
1560          '035 016 015'
1561          '135 776 000'
1562 UINVM345(7)                                    " RM5 STACK GRANTED
1563          '031 026 135'
```

```
1564          '005 135 034'
1565          '035 012 014'
1566          '024 135 020'
1567          '033 012 027'
1568          '035 016 015'
1569          '135 776 000'

1570 C1NVMNN(2)
1571          '027 027 135'                    " NN
1572          '776 000 000'
1573 C1NVM13(3)
1574          '027 030 035'                    " NOT OUT.
1575          '135 030 036'
1576          '035 130 776'
1577 C1NVM9(8)
1578          '012 025 033'                    " ALREADY DISCONNECTED.
1579          '016 012 015'
1580          '042 135 015'
1581          '022 034 014'
1582          '030 027 027'
1583          '016 014 035'
1584          '016 015 130'
1585          '776 000 000'
1586 C1NVM7(5)
1587          '012 034 024'                    " ASKED ALREADY.
1588          '016 015 135'
1589          '012 025 033'
1590          '016 012 015'
1591          '042 130 776'
1592 C1NVM6(5)
1593          '027 030 135'                    " NO RM EMPTY.
1594          '031 026 135'
1595          '016 026 031'
1596          '035 042 130'
1597          '776 000 000'
1598 U1NVM12A(7)
1599          '033 016 135'                    " RE A DISCONNECTED.
1600          '012 135 015'
1601          '022 034 014'
1602          '030 027 027'
1603          '016 014 035'
1604          '016 015 130'
1605          '776 000 000'
1606 U1NVM12B(7)
1607          '033 016 135'                    " RE B DISCONNECTED.
1608          '013 135 015'
1609          '022 034 014'
1610          '030 027 027'
1611          '016 014 035'
1612          '016 015 130'
1613          '776 000 000'
1614 U1NVM12C(7)
1615          '033 016 135'                    " RE C DISCONNECTED.
1616          '014 135 015'
1617          '022 034 014'
1618          '030 027 027'
1619          '016 014 035'
1620          '016 015 130'
1621          '76 000 000'
1622 U1NVM8A(4)
```

```
1623          '033 016 135'                    " RE A FREE.
1624          '012 135 017'
1625          '033 016 016'
1626          '130 776 000'
1627 UINVM8B(4)
1628          '033 016 135'                    " RE B FREE.
1629          '013 135 017'
1630          '033 016 016'
1631          '130 776 000'
1632 UINVM14C(8)
1633          '033 016 026'
1634          '030 037 016'
1635          '135 035 012'
1636          '031 016 135'
1637          '017 033 030'
1638          '026 135 033'
1639          '016 135 014'
1640          '174 776 000'
1641 UINVM14A(8)
1642          '033 016 026'                    " REMOVE TAPE FROM RE A+
1643          '030 037 016'
1644          '135 035 012'
1645          '031 016 135'
1646          '017 033 030'
1647          '026 135 033'
1648          '016 135 012'
1649          '174 776 000'
1650 UINVM14B(8)                               " REMOVE TAPE FROM RE B+
1651          '033 016 026'
1652          '030 037 016'
1653          '135 035 012'
1654          '031 016 135'
1655          '017 033 030'
1656          '026 135 033'
1657          '016 135 013'
1658          '174 776 000'
1659 UINVM1A(2)
1660          '033 016 135'                    " RE A
1661          '012 135 776'
1662 UINVM1B(2)
1663          '033 016 135'                    " RE B
1664          '013 135 776'

1665 UINVM1C(2)
1666          '033 016 135'                    " RE C
1667          '014 135 776'
```

```
 0 CLPPROC2(1)
 1          CLPPROC2
 2 CABSFIXT(2)
 3          $ = - 0
 4          JUMP(3)
 5 CFIXT(2)
 6          $ = 0
 7          JUMP(1)
 8 CFLOT(42)
 9          $ = 1
10          CGL2 = $
11          A = 1
12          $ = :MA[3]
13          SE18                          " CREATE DISPLAY
14          $ = MA[- 6]
15     U,   $ - 21, Z                     " THREE PARAMETERS?
16     N,   SUBCD(:RSE19)                 " ALARM
17          F = :MA[- 11]
18          SUBCD(:RSE20)                 " CHECK FORMAL ARITHMETIC
19          F = :MA[- 15]
20          SUBCD(:RSE20)
21          F = :MA[- 19]
22          SUBCD(:RSE20)
23          $ = 40                        " ANONYMOUS WORKING SPACE
24          SE1
25          $ = CGL2                      " RETRIEVE
26          MC = $                        " AND STACK FIX/FLO INDICATION
27          DOS(MA[- 19])
28          DO(ERV)                       " EVALUATE N
29     U,   $ = F, Z
30     N,   SUBCD(:RSE88)                 " ROUND N
31          MC = G                        " STACK N
32          DOS(MA[- 15])
33          DO(ERV)                       " EVALUATE M
34     U,   $ = F, Z
35     N,   SUBCD(:RSE88)                 " ROUND M
36          MC = G                        " STACK M

37          DOS(MA[- 11])
38          DO(ERV)                       " EVALUATE X
39          MC = F                        " STACK X
40          SUBC(:MT[144])
41          SUBC(:MT[207])
42          $ = ULRSTREAMS
43          $ = MS
44          UBLOCKSTR = $
45     U,   A = MS[6], R
46     Y,   JUMP(2)
47          CHEEN
48          CLPSI
49          SE33                          " SEGMENT TRANSITION
50          CONNUM                        " CONTINUATION PRINT NUMBER
51 CRRNTX(28)
52          A = 1
53          $ = :MA[3]                    " ONE PARAMETER?
54          SE18
55          $ = MA[- 6]
56     U,   $ - 13, Z                     " CHECK FORMAL
57     N,   SUBCD(:RSE19)
```

dec numbers printing procedures

```
58          F = :MA[- 11]
59          $UBCD(:RSE20)
60          $ = 40                          " EVALUATE X
61          SE1                             " TO MAKE SPACE
62          DOS(MA[- 11])
63          DO(ERV)
64          B + 3
65      U,  INT                             " NINT := 0
66          MC = F
67          INT
68          SUBC(:MT(117))

69          SUBC(:MT[1])
70          JUMP(- 28)                      " TO MERGE WITH CELOT
71          $ = MC[- 1]
72          MG[- 1] = $
73      N,  JUMP(185)
74          $ = 0
75          MG[ - 4] = $
76          $ = 13
77          MG[ - 5] = $
78          MG[ - 6] = $
79          JUMP(237)                       " TO FIX

80  CRNCH2(2)
81          F = 2
82          JUMP(1)
83  CRNCH1(36)          number punching
84          F = 1
85          CGL1 = G
86          A = 1
87          $ = :MA[3]
88          SE18
89          $ = 13
90      U,  $ - MA[ - 6], Z                 " MERGING POINT FROM CRNCH1
91      N,  $UBCD(:RSE19)
92          $ - 13, P                       " TWO PARAMETERS ?
93      Y,  F = :MA[ - 15]
94      Y,  $UBCD(:RSE20)                   " CHECK I
95          F = :MA[ - 11]
96          $UBCD(:RSE20)                   " CHECK X
97          $ = 40
98          SE1
99      N,  $ = CGL1
100         MC = $                          " SAVE STREAMNUMBER
101     N,  JUMP(9)
102         DOS(MA[ - 15])
103         DO(ERV)
104     U,  $ = F, Z
105     N,  $UBCD(:RSE88)                   " ROUND STREAMNUMBER
106         $ = G
107     U,  $ - UNUMTRRMSTR, P
108     N,  $ - 0, E
109     Y,  JUMP(353)                       " TO FLOR1, ALARM WRONG STREAMNUMBER
110         M[B - 1] = $                    " FILL IN STREAMNUMBER
111         DOS(MA[ - 11])
112         DO(ERV)
113         B + 3
114     U,  INT
115         MC = F
```

```
104 - 262 -   200669 - 01112562 SYSTEEMPRI
116           INT
117           SUBC(:MT[70])                    " CONVRT
118           SUBC(:MT[ - 46])                 " CRRNTX[19]
119           JUMP(361)                        " TO MERGE WITH CELOR
120 CRNCH1(5)
121           A = 1
122           S = :MA[3]
123           SE18
124           S = 17
125           JUMP( - 35)                      " TO MERGE WITH CRNCH1

126 CPAGEHEADING(59)   takes case of this
127           S = :MG[16]
128           MG[26] = S
129           LCS(9)
130           S + 1
131           MG[21] = S
132           S = MG[14]
133           MC = S                           " SAVE LINEFEED SPECIFICATION
134           S = 1
135           MG[14] = S
136           MG[13] - S                       " PRECORRECTION OF LINENUMBER
137           S = 0
138           MG[10] = S                       " DRUK := TRUE
139           S = URUNNUM
140           SUBC(:MT[31])
141           CHEEN
142           CHEADN
143           S = 1
144           G = UBLOCKSTR
145           RLUSS(MG[15])                    " PAGENUMBER
146     U,    S - 999, 9
147     Y,    S = - 0
148     Y,    MG[15] = S
149     U,    S '*' 31, Z                      " 32 FORMS PRODUCED ?
150     N,    JUMP(12)
151           MC = S
152           SUBCD(:UIMMORTAL)
153           SUBCD(:UCLAIMCONRED)
154           S = :M0[0]
155           SUBCD(:UPROGNAME)
156           A = M[B - 1]
157           SUBCD(:CRRINTDEC)
158           G = MT[12]
159           SUBC(:CRRINTSTRING)
160           SUBCD(:UCONEREEVHITE)
161           SUBCD(:UMORTAL)
162           S = MC[- 1]

163           A = 3
164           SUBC(:MT[8])
165           CHEEN
166           CHEADN
167           S = UDATE
168           SUBC(:MT[3])
169           SE33
170           CHEADS
171           CINVFORMMES
172           A = 8                            " SETUP
173           M[B + 1] = A                     " FILL N
```

```
174           A = 0
175           B + 5
176           G = S
177           M[B - 5] = - A                  " ABSFIXT
178           M[B - 3] = A                    " N := 0
179           M[B - 2] = E
180           SUBC(:MT[9])                    " CONVRT
181           SUBC(:MT[1])
182           GOTOR(MG[- 7])
183           S = MC[- 1]
184           MG[- 1] = S
185           JUMP(135)
186 CRUV!$NUM(4)
187           SUBC(:MT[ - 14])                " PREPARE NUMBER
188           S = MG[ - 8]                    " RETRIEVE INVARIANT LINK
189           MC = S                          " AND STACK FOR RETURN
190           CTERUG

191 CONVRT(64)                                                    bin-dec. conversion
192           E = M[B - 3], P                 " E := X
193       N,  E = - E                         " E := ABS(X)
194           A = - E
195           S = E
196           RUS(15)                         " BINARY EXPONENT IN S
197           M[B + 3] = S                    " SAVE BINARY EXPONENT
198           A '*' 32 767
199           A '+' - 32 767
200           S = - G                         " - MANTISSA IN A$
201           G = B                           " G TO BE USED AS WORKING SPACE POINTER
202           NORAS                           " NORMALISE MANTISSA
203           B - 52
204           MG[3] - B                       " BINARY EXPONENT FOR NORMALISED MANTISSA
205           MG[1] = A                       " HEAD := A
206           A = - 0
207           MG[0] = A                       " INITIALISE DECIMAL EXPONENT TO - 0
208           B = 10
209           PLUSA(MG[3]), P                 " UPDATE BEXP, MORE DIVIDING TO DO ? ( :DIV)
210       N,  JUMP(21)                        " GOTO(:MULT)
211           MG[0] - B                       " UPDATE DEXP
212           A = MG[1]
213       U,  A + CTENPO[B - 1], P            " CAN WE DIVIDE WITHOUT OVERFLOW ?
214       N,  RUAS(1)                         " PRESHIFT FOR DIVISION
215           DIVAS(CTENPO[B - 1])
216           MG[1] = S                       " STORE HEAD
217           DIVA(CTENPO[B - 1])             " DIVIDE LESS SIGNIFICANT HALF
218           A = -CSCALE[B - 1]
219       N,  A + 1                           " CORRECTION FOR SHIFT
220           JUMP( - 12)                     " GOTO(:DIV)
221           A + CSCALE[B - 1]               " UPDATE BEXP  (:MUL)
222           MG[3] = A
223           MG[0] + B
224           MULS(CTENPO[B - 1])
225           S = MG[1]
226           MULAS(CTENPO[B - 1])
227           LCAS(1), P                      " NORMALISE, WAS IT ALREADY NORMALISED ?
228       Y,  RCAS(1)                         " THEN RESTORE
229           MG[1] = A
230           A = MG[3]
231       N,  A - 1                           " CORRECT FOR NORMALISING SHIFT
232       U,  A + CSCALE[B - 1], P            " DONE WITH MULTIPLICATIONS ?  (:MULT)
```

```
233     N,  JUMP( - 13)                 " GOTO(:MUL)
234         B - 1, P                    " MORE MULTIPLICATIONS POSSIBLE ?
235     Y,  JUMP( - 4)
236         B = - A                     " REMAINS OF BINARY EXPONENT
237     U,  B - 3, Z                    " IS IT POSSIBLE TO MULTIPLY BY 10 / 16
238     N,  JUMP(8)                     " GOTO(:AH)
239         MG[2] = S                   " TAIL := $
240         MUL$(CTENPO)
241         S = MG[1]
242         MULA$(CTENPO)
243         LCAS(1), E                  " MULTIPLY BY 2, WAS THAT OK ?
244     Y,  B = 1
245     Y,  MG[0] + B                   " UPDATE DEXP
246     N,  S = MG[2]
247     N,  A = MG[1]
248     N,  RUA$(B)                     " NORMALISE DECIMAL MANTISSA
249         MG[1] = - A                 " STORE POSITIVE
250         MG[2] = - S                 " DECIMAL MANTISSA
251         S = - MG[0], Z
252     Y,  S = 0                       " DEXP := + 0
253         MG[0] = S                   " STORE DEXP WITH CORRECT SIGN
254         B = :MG[4]                  " TO BE CONTINUED BY CFIXFLO
255         GOTOR(MG[ - 1])

256 CFIXFLO(166)                        create row of digits and . + - 10 "blank" on stack according to layout specifications.
257         S = MC[ - 1]
258         MG[ - 1] = S                " STORE LINK
259         S = - 0
260         S + MG[ - 6], Z
261     Y,  JUMP(55)                    " TO TEST FIX
262     U,  S - MG[ - 4], P             " M ≤ 0
263     N,  S - MG[ - 5], P             " N ≤ 0
264     N,  S + 13, E                   " N > 13
265     N,  JUMP(7)                     " TO FLO
266         S = 1, Z                    " SPILL
267         MG[ - 6] = S                " MAKE FLOT (OR FIX, MERGING POINT FROM PRINT(X))
268     N,  S = 3                       " M := 3
269         MG[ - 4] = S
270         S = 13
271         MG[ - 5] = S                " N := 13
272     Y,  JUMP(51)                    " TO FIX

273         S = 0
274         B + 1                       " SKIP SIGN POSITION, FLO
275         MC = S                      " 0 IN POSITION OF DECIMAL POINT
276         S = MG[ - 5]                " NUMBER OF MANTISSA DIGITS
277         S + 1                       " ONE EXTRA DIGIT
278         SUBC(:MT[112])              " CMND
279         S - 1, Z
280     N,  MC = A
281     N,  JUMP( - 4)
282         MG[3] = B                   " SAVE END OF DIGITS POINTER
283         SUBCD(:MT[116])             " CROUND
284         S = MG[5], Z                " OVERFLOW OF MANTISSA BY ROUNDING ?
285     N,  MG[6] = S                   " SHIFT 1 TO NEXT POSITION
286         PLUS$(MG[0]), P             " INCREASE DEXP
287         B = MG[3]
288         A = 13                      " 10
289         MC = A
290     N,  S = - S, Z
```

```
291     N,  A = 11                          " -
292     Y,  A = 10                          " +
293         MC = A
294         A = 12
295         MG[5] = A                       " PLACE DECIMAL POINT
296         A = MG[ - 4]
297     U,  A - 3, P                        " M > 3
298     Y,  JUMP( + 85)
299         A + :CTAB9[- 1]                 " 9, 99, 999
300     U,  $ - MA, P                       " DEXP > 10 ↑ M - 1 ?
301     Y,  JUMP(82)                        " TO REPEAT CONVERSION
302         A = 1
303         MINA(MG[ - 4]), Z               " DESTROYING M
304     Y,  MC = $
305     Y,  JUMP(70)                        " TO ADD SIGN AND TRAILING SPACE
306     U,  A - 1, Z
307         A = 0
308     Y,  DIVA$(10)
309     N,  DIVA$(100)
310         MC = $ , Z                      " RESULT DIGIT
311     Y,  $ = M[B - 2]                    " PREVIOUS SYMBOL
312     Y,  $ - 9, P                        " - DIGIT
313     Y,  $ = 14                          " BLANK
314     Y,  M[B - 1] = $                    " TRANSFORM LEADING 0 TO BLANK
315         $ = A
316         JUMP( - 15)

317     U,  $ - MG[ - 4], P                 " TEST FIX, $ = - 0, M < 0 ?
318     N,  $ - MG[ - 5], P                 " N < 0 ?
319     N,  $ + MG, P                       " DEXP > N ?
320     N,  $ - MG
321     N,  $ - MG[ - 4], Z                 " M + N = 0 ?
322     N,  $ + 22, E                       " N + M > 21
323     Y,  JUMP( - 58)                     " TO SPILL
324         $ = MG, P                       " DEXP ≥ 0 ?
325     N,  $ + MG[ - 5]
326     Y,  $ = MG[ - 5]
327         A = 14                          " BLANK
328         MC = A                          " PRODUCE LEADING BLANKS
329     U,  $ - MG, Z
330     N,  $ - 1
331     N,  JUMP( - 4)
332         A = 0
333         M[B - 1] = A                    " REPLACE LAST BLANK BY 0
334         $ - 0, P                        " NUMBER OF DIGITS IN FRONT OF POINT > 0 ?
335     Y,  SUBC(:MT[55])                   " CRND
336     Y,  MC = A
337     Y,  $ - 1, P
338     Y,  JUMP( - 4)
339         $ = MG[ - 4], Z                 " M = 0 ?
340     N,  A = 9
341     N,  MC = A                          " 9 ON FUTURE POSITION OF DECIMAL POINT
342         A = 0
343         $ + MG, P                       " M + DEXP > 0
344     N,  $ + 0, Z                        " M + DEXP = 0
345     N,  JUMP(11)
346         $ + 1
347     U,  $ - MG[ - 4], P
348     N,  MC = A
349     N,  JUMP( - 4)
```

flo = floating — n digits mantissa, m digits exponent ≤ 3
fix = fixed — n digits before point, m digits after point

→ to insert standard format flo m=13 m=3

```
350     U,  S = MG, P
351     Y,  S - MG
352     N,  S + MG
353         SUBC(:MT[37])                   " CRND
354         S - 1, Z
355     N,  MC = A
356     N,  JUMP( - 4)
357     N,  S = MG[ - 4], Z                 " PRODUCE M ZEROES
358     N,  MC = A
359     N,  S - 1, Z
360     N,  JUMP( - 3)
361         MG[3] = B
362         SUBCD(:MT[37])                  " UPON RETURN S = - 0
363         B - :MG[4], Z                   " OVERFLOW INTO SIGNPOSITION ?
364         B = :MG[4]
365     Y,  JUMP(18)

366         B + MG[ - 5]                    " + N
367         A = 12                          " DECIMAL POINT
368         M[B + 1] = A                    " IRRELEVANT WHEN M = 0
369     U,  S + MG, P                       " DEXP > 0
370     Y,  B - MG                          " B TO POINT TO LEADING ZERO
371     N,  S + MG[ - 4], P                 " LEADING ZERO IN UNITS POSITION ^ M ≠ 0
372     Y,  A = M[B], Z
373     Y,  A = 14
374     Y,  M[B] = A
375         B = MG[3]
376         A = 14
377         MC = A                          " ADD TRAILING SPACE
378         MG[4] = A                       " AND LEADING SPACE
379         S = MG[ - 6], P                 " ~ ABSFIXT ?
380     N,  GOTOR(MG[ - 1])
381         S - 7, P                        " PRINT(INTEGER) ?
382     N,  JUMP(31)
383     Y,  JUMP(27)
384         B = :MG[ - 1]
385         S = M[B]                        " SAVE LINK OF CALL OF CFIXFLO
386         MG[4] = S
387         SUBC(:MT[ - 1951)               " REPEAT CONVERSION
388         S = MG[4]
389         MG[ - 1] = S                    " RESTORE LINK
390         JUMP( - 125)                    " TO SPILL

391 "CRND(9)
392         MG[3] = S                       " SAVE S IN WORKING SPACE
393         S = MG[2]                       " TAIL
394         TENAS
395         MG[2] = S
396         S = MG[1]                       " HEAD
397         MULAS(10)                       " NEXT DIGIT IN A.
398         MG[1] = S
399         S = MG[3]
400         GOTOR(MC[ - 1])
401 "CROUND(11)
402         S = MC[ - 1]                    " UNSTACK LINK
403         UWS[4] = S                      " AND SAVE
404         S = - 0
405         A + 5
406     U,  A - 9, P                        " DIGIT OVERFLOW ?
407     N,  GOTOR(UWS[4])
```

call of CROUND (relative addressing)

another working space in core, poorly making use of deafness in call of CROUND

```
408             A = 1
409             M[B] = - $
410             A + MC[ - 1]                    " INCREASE PREVIOUS DIGIT, DECREASING B
411             M[B] = A
412             JUMP( - 7)

413             MC = A                          " ADD 6 MORE BLANKS
414             $ - 1, P
415         Y,  JUMP( - 3)
416             $ = :MG[5]                      " FIND SIGN POSITION
417         U,  A - MS, Z                       " BLANK IN THIS POSITION ?
418         Y,  $ + 1
419         Y,  JUMP( - 3)
420         U,  A = MG[ - 2], P
421         Y,  A = 10
422         N,  A = 11
423             MS[- 1] = A                     " PLACE SIGN
424             GOTOR(MG[ - 1])

425 CABSFIXR2(2)
426             F = 2
427             JUMP(1)
428 CABSFIXR1(3)
429             F = 1
430             $ = - 0
431             JUMP(9)
432 CFIXR2(2)
433             F = 2
434             JUMP(1)
435 CFIXR1(3)
436             F = 1
437             $ = 0
438             JUMP(4)
439 CFLOR2(2)
440             F = 2
441             JUMP(1)
442 CFLOR1(63)
443             F = 1
444             $ = 1
445             CGL1 = G                        " SAVE STREAMNUMBER AND FIX/FLO INDICATION
446             CGL2 = $
447             A = 1
448             $ = :MA[3]
449             $E18
450             $ = 21
451         U,  $ - MA[ - 6], Z                 " MERGING POINT
452         N,  SUBCD(:RSE19)
453             $ - 21, P
454         Y,  F = :MA[ - 23]
455         Y,  SUBCD(:RSE20)                   " CHECK I
456             F = :MA[ - 19]
457             SUBCD(:RSE20)                   " CHECK N
458             F = :MA[ - 15]
459             SUBCD(:RSE20)                   " CHECK M
460             F = :MA[ - 11]
461             SUBCD(:RSE20)                   " CHECK X
462             $ = 40
463             $E1
464         N,  $ = CGL1                        " RETRIEVE AND STACK
465             MC = $                          " STREAMNUMBER
```

more numbers printing & punching procedures

```
466             $ = CGL2
467             MC = $                          " AND FIX/FLO INDICATION
468         N,  JUMP(10)
469             DOS(MA[ - 23])                  " EVALUATE STREAMNUMBER
470             DO(ERV)
471         U,  $ = E, Z
472         N,  SUBCD(:RSE88)
473             $ = G
474         U,  $ = UNUMTRMSTR, P
475         N,  $ = 0, E
476         Y,  $ = 4
477         Y,  SUBCD(:CALARM)                  " NON EXISTENT STREAMNUMBER
478             M[B - 2] = $                    " FILL STREAMNUMBER
479             DOS(MA[ - 19])
480             DO(ERV)
481         U,  $ = E, Z                        " EVALUATE AND
482         N,  SUBCD(:RSE88)                   " ROUND N
483             MC = G
484             DOS(MA[ - 15])
485             DO(ERV)                         " EVALUATE AND
486         U,  $ = E, Z
487         N,  SUBCD(:RSE88)                   " M
488             MC = G
489             DOS(MA[ - 11])
490             DO(ERV)                         " EVALUATE X
491             MC = E
492             SUBC(: MT[12])                  " CONVRT VIA JUMP
493             SUBC(:MT[ - 229])               " CFIXFLO
494             $ = UTRSTREAMS                  " MERGING POINT FROM CRNCM
495             $ + MG[ - 7]                    " ADD STREAMNUMBER
496             $ = MS[ - 1]
497             UBLOCKSTR = $
498         U,  A = MS[6], P                    " DOCUMENT UNDER CONSTRUCTION ?
499             SUBCD(:UIMMORTAL)
500         Y,  JUMP(2)
501             CHEEN
502             CBEGINTRDOC                     " TO LEAVE G UNALTERED
503             SE33
504             CONRUNUM
505             JUMP( - 305)                    " TO CONVRT

506 CABSE!XR!(2)
507             $ = - 0
508             JUMP(3)
509 CF!XR!(2)
510             $ = 0
511             JUMP(1)
512 CFLOR!(7)
513             $ = 1
514             CGL2 = $
515             A = 1
516             $ = :MA[3]
517             SE18
518             $ = 25
519             JUMP( - 66)                     " TO MERGE WITH CFLOR1

520 CLRS!(15)
521             A = - 0
522             MS[12] = A
523             MS[13] = A
```

```
524           MS[14] = A
525           MS[16] = A
526           MS[17] = A
527           MS[18] = A
528           MS[19] = A
529           MS[20] = A
530           MS[24] = A
531           MS[10] = A
532           A = - 1
533           MS[15] = A
534           MS[6] = - A                    " DUC := IRUE
535           CTERUG
536 CINVFORMMES(8)
537           '031 033 022'                  " RRI
538           '027 035 016'                  " NTE
539           '033 135 017'                  " R E
540           '030 033 026'                  " ORM
541           '034 135 031'                  " $ R
542           '033 030 015'                  " ROD
543           '036 014 016'                  " UCE
544           '015 130 776'                  " D.

545 CLRRROC1(54)
546           CLRRROC1
547           S = 0                          " CONVINTLR(37)
548       U,  A - 383, R                     code conversion
549       U,  A + 1, E
550       Y,  A = 2                          " VALUE OUT OF RANGE
551       Y,  GOTOR(MC[- 1])
552       U,  A - 9, R                       " - DIGIT
553       N,  A + 16
554       N,  GOTOR(MC[- 1])
555       U,  A - 35, R                      " - LETTER
556       N,  A + 23
557       N,  GOTOR(MC[- 1])
558       U,  A - 36, Z
559       Y,  A + 27
560       U,  A - 62, R                      " - CAPITAL LETTER?
561       N,  S = 14                         " FOR .
562       N,  A - 4
563       N,  GOTOR(MC[- 1])
564       U,  A - 93, Z                      " BLANK SPACE(SPEED UP)
565       Y,  A = 0
566       Y,  GOTOR(MC[- 1])
567       U,  A - 129, R
568       Y,  JUMR(4)
569           A = :MA[- 63], Z
570       Y,  S = 0
571           SUBC(:MT[9])
572           GOTOR(MC[- 1])
573       U,  A - 255, R                     " SHORTCUT FROM CRRSYM
574       Y,  S = 28                         " STROKE
575       N,  S = 32                         " UNDERLINING
576           A '*' 127
577       U,  A - 36, Z
578       U,  A - 62, E
579       N,  JUMR(- 18)
580           JUMR(- 29)
581           SUBCD(:UBEDDEN)
582           E = :CTABINTLR
```

line printer procedures

capital letters are printed as A
small " " " " A

```
583          GOTO(:UCONVTAIL)

584          S = MC[- 1]                    " CKOPSTRING(16)
585          CGL2 = S
586          A = 1
587          S = :MA[3]
588          SE18
589          S = MA[- 6]
590      U,  S - 13, Z                       " ONE PARAMETER?
591      N,  SUBCD(:PSE19)
592          F = :MA[- 11]
593          SUBCD(:PSE29)                  " CHECK FORMAL STRING
594          S = 25                         " ANONYMONS WORKING SPACE
595          SE1
596          G = MA[- 9]
597          S = CGL2                       " STACK LINK
598          MC = S                         " GET INVARIANT ADDRESS FROM F[2]
599          JUMP(82)                       " TO MERGING POINT
600 CPRINTTEXT(22)
601          SUBC(:MT[- 17])
602          SUBC(:MT[1])
603          JUMP(43)                       " TO EXIT"
604          SUBCD(:UINSTV)
605          F = :MC[ - 3]
606          SUBCD(:UDNSS)
607      U,  A - 510, Z                      " END OF STRING?
608      Y,  B - 3                          " KILL STRING VARIABLE
609      Y,  JUMP(80)                       " TO RETURN VIA LINK
610          SUBC(:MT[88])                  " CPRSYM + 1
611          JUMP(- 7)

612          S = MC[- 1]                    " CKOP(9)
613          CGL2 = S
614          A = 1
615          S = :MA[3]
616          SE18
617          S = MA[- 6]
618      U,  S - 9, Z                        " NO PARAMETERS?
619      N,  SUBCD(:PSE19)
620          S = 20
621          SE1                            " ANONYMOUS WORKING SPACE
622          JUMP(- 25)                     " TO MERGE WITH CKOPSTRING
623 CHEADN(8)
624          S = MC[ - 1]                   " UNSTACK INVARIANT LINK
625          MG[ - 7] = S
626          SUBC(:MT[157])
627          F = 13
628          MC = F
629          SUBC(:MT[237])
630          B - 2
631          CTERUG
632 CTAB(15)
633          SUBC(:MT[- 20])
634          A = MS[12]
635      U,  A - 134, P
636      N,  JUMP(5)
637      U,  A - 142, P
638          SUBC(:MT[193])
639          F = - 0
640      Y,  F = 8
```

```
641           JUMP(202)                       " MERGE WITH SPACE(N)
642           A '*' 7
643       U,  A - 7, Z
644       Y,  A = - 1
645           F = B
646           G - A
647           JUMP(196)                       " TO MERGE WITH SPACE(N)

648 CONNUM(6)
649           SUBC(:MT[136])
650           S = 100
651           GBILL + S, P                    " EXIT
652           SE115
653           DO(MD[- 1])
654           SE71
655 CENDLPDOC(8)
656           S = UBLOCKSTR
657           SUBC(:MT[156])
658       U,  S = MS[24], P                   " AE?
659       Y,  F = - 1
660       Y,  SUBCD(:CAANLP)
661       Y,  S = G
662           MS[6] = - B
663           CTERUG
664 CNEWPAGE(34)
665           SUBC(:MT[ - 49])
666       Y,  SUBC(:MT[148])
667           S = 20
668           JUMP( - 16)
669           0

670           S = MC[- 1]                     " CKORN(29)
671           CGL2 = S                        " STORE LINK IN GLOBAL
672           A = 1
673           S = :MA[3]
674           SE18
675           S = MA[- 6]
676       U,  S - 13, Z                       " ONE PARAMETER?
677       N,  SUBCD(:RSE19)
678           F = :MA[- 11]
679           SUBCD(:RSE20)                   " CHECK FORMAL ARITHMETIC
680           S = 20
681           SE1
682           S = CGL2
683           MC = S                          " STACK LINK
684           DOS(MA[- 11])
685           DO(ERV)
686       U,  S = F, Z
687       N,  SUBCD(:RSE88)
688           S = ULPSTREAMS                  " MERGING POINT
689           S = MS
690           UBLOCKSTR = S
691       U,  A = MS[6], P                    " DOCUMENT UNDER CONSTRUCTION?
692       Y,  JUMP(2)
693           CHEEN
694           CLRSI                           " INITIALISE STREAM
695           A = MC[- 1]                     " RETURN VIA LINK
696           A '*' 511                       " COLLATE OUT LINENUMBER
697           A - 147
698           JUMP(A)
```

```
699 CPRSYM(50)
700          SUBC(:MT[- 30])
701          SUBC(:MT[2])
702          S = 25
703          JUMP(- 50)                    " TO EXIT
704          A = G                         " CPRSYM
705     U,   A - 119, Z
706     Y,   SUBC(:MT[129])
707     Y,   GOTOR(MC[- 1])
708          SUBC(:MT[- 155])              " CONVINTLP
709     U,   S - 27, P                     " S = 28 V S = 32?
710     Y,   A = 2
711     Y,   S = 0                         " TO PRINT NON EXISTENT VALUE
712     U,   A - 63, P
713     Y,   JUMP(5)
714          MC = S
715          MC = A
716          SUBC(:MT[155])
717          B - 2
718          JUMP(- 23)                    " TO RETURN VIA LINK
719     U,   A - 127, P                    " SYM GIVEN AS CHARF-VALUE?
720     Y,   SUBC(:MT[- 141])              " SHORTCUT
721     Y,   JUMP(- 8)
722          G = UBLOCKSTR
723          S = MG[12]
724     U,   S - 134, P                    " - ROOM FOR WORDDELIMITER
725     Y,   MC = A
726     Y,   SUBC(:MT[109])
727     Y,   A = MC[ - 1]
728          B + 4                         " MAKE ROOM FOR LOCALS
729          S = 32                        " UNDERLINING
730          M[B - 2] = S                  " STACK ONCE AND FOR ALL
731          A + !CTABWDL[- 64]
732          M[B - 3] = A                  " SAVE LISTPOINTER
733          S = MA
734          A = 31
735          A '*' S, Z                    " COLLATE OUT ONE LETTER DONE?
736     Y,   JUMP(11)

737          RUS(5)                              " PREPARE FOR NEXT LETTER
738          M[B - 4] = S
739          A + 32
740          M[B - 1] = A                  " STACK FOR CALL OF CURRNT
741          SUBC(:MT[130])
742          S = M[B - 4], Z
743     N,   JUMP(- 10)
744          S + 0, P                      " NO WORD TO FOLLOW
745     N,   A = M[B - 3]                  " RECALL LIST POINTER
746     N,   S = MA[1]
747     N,   JUMP(- 14)
748          B - 4
749          JUMP(- 54)                    " RETURN VIA LINK

750 CNLCR(3)
751          SUBC(:MT[ - 133])
752     Y,   SUBC(:MT[84])
753          JUMP( - 85)                   " TO EXIT
754 CHEADS(15)
755          SUBC(:MT[35])
```

```
756             F = 13
757             MC = F
758             SUBC(:MT[115])
759             M[B - 1] = S                    " UPON RETURN FROM CLPRNT S = 0
760             SUBC(:MT[113])
761             B = 3
762             G = UPROGID
763             SUBC(:MT[ - 152])               " PRINTTEXT
764             SUBC(:MT[73])
765             G = S
766             S = 2
767             S + MC[ - 1]
768             MG[14] = S                      " RESTORE LINEFEED
769             JUMP(120)                       " TO CLPRNT[16]

770 CPRCHAR(78)
771             SUBC(:MT[- 98])
772             A = G
773         U,  A - 119, Z
774         Y,  SUBC(:MT[64])
775         Y,  JUMP( - 105)
776             SUBC(:MT[- 220])
777         U,  A - 63, P
778         Y,  JUMP(5)
779             MC = S
780             MC = A                          " MAY BE SHORTENED INTO
781             SUBC(:MT[93])                   " CALL OF SUBC(:MT[CPRSYM + 10])
782             B - 2
783             JUMP(- 113)                     " TO EXIT
784         U,  A - 215, Z                      " A = 1
785         Y,  A = 1                           " FOR DIGIT 1
786         N,  A - 99, Z                       " SYM-VALUE FOR NLCR
787         Y,  A + 16                          " FOR DIGIT 0
788         Y,  S = 32
789         N,  S = 0
790         N,  A = 2
791             JUMP(- 13)

792             S = MC[- 1]                     " CPRNUM(27)
793             MG[- 6] = S                     " STORE LINK, OVERWRITING FIX/FLO INDIC,
794             MC = G                          " SAVE G
795             S = :MG[4]
796             A = :MC[- 145]
797             G = UBLOCKSTR
798             A + MG[12]
799             A - S, P                        " SPACE ON LINE INSUFFICIENT?
800             MC = S                          " STORE POINTER
801         Y,  SUBC(:MT[37])
802         Y,  S = M[B - 1]
803             F = 0
804             MC = F
805             A = MS
806             A - 9, P                        " - DIGIT
807         N,  A + 25
808         Y,  A + :CNUMLPTAB[- 1]
809         Y,  A = MA
810             M[B - 1] = A
811             SUBC(:MT[63])
812             S = 1
813             PLUSS(M[B - 31])
```

continuation CPAGEHEADING

```
814     U,  S - :MC[- 5], P             " S POINTING PAST LAST SYMBOL?
815     N,  JUMP(- 11)
816         B = M[B - 4]                " FORMER VALUE OF G
817         B - 5
818         JUMP(- 120)                 " TO RETURN VIA LINK

819     U,  S = MS[12], Z               " CNEWPAGE(20)
820     N,  SUBC(:MT[172])
821     Y,  A = - MS[10], P
822     Y,  JUMP(12)
823         S = UBLOCKSTR
824         A = MS[26]
825         A - :MS[15]
826         G = A
827         S = :MS[16]
828         LCS(9)                      " MAKE INVARIANT ADDRESS
829         MC = S
830         SUBCD(:PSE50)               " FILL FIRST WORD OF FIRST SEGMENT
831         A = 0
832         S = UBLOCKSTR
833         MS[24] = A                  " AF := IBUE
834         MS[10] = - A                " DBUK := EALSE
835         MS[12] = - A
836         MS[13] = A
837         MS[14] = A
838         GOTOR(MC[- 1])
839         S = UBLOCKSTR               " CNLCR(10)
840     U,  S = MS[12], Z
841     N,  SUBC(:MT[151])
842     N,  S = UBLOCKSTR
843         A = 1
844         MS[14] + A
845         PLUSA(MS[13])
846     U,  A - 60, Z
847     Y,  SUBC(:MT[- 29])
848         GOTOR(MC[- 1])

849 CSPACE(13)
850         SUBC(:MT[- 176])
851         F - 0
852         MC = G                      " STACK NUMBER OF SPACES
853         MC = G, P                   " TO BE GIVEN TWICE
854         F = 0
855         MC = F                      " STACK NON UNDERLINED SPACE
856     Y,  SUBC(:MT[19])
857         S = 1
858         M[B - 3] - S, P
859     Y,  JUMP(- 4)
860         B - 3
861         S = MC[- 1]                 " PRICE = NUMBER OF SPACE/NLCRS
862         JUMP(- 205)                 " TO EXIT
863 CARRIAGE(10)
864         SUBC(:MT[- 189])
865         F - 0
866         MC = G
867         MC = G, P
868     Y,  SUBC(:MT[- 28])
869         S = 1
870         M[B - 1] - S, P
871     Y,  JUMP(- 4)
```

```
872          B - 1
873          JUMP(- 12)

874 GLPRNT(189)          → basic routine to print one character including underlining, strokeing or "capital/period"
875          G = UBLOCKSTR
876          $ = MG[12], Z                    " POS > 0
877      N,  $ - 143, E                       " ^ POS < 144
878      N,  JUMP(44)
879          $ + 0, Z                         " WAS POS = 0?
880      Y,  JUMP(2)
881          SUBC(:MT[- 43])
882          G = $                            " RESTORE UBLOCKSTR IN G
883          $ = MG[24], P
884      Y,  F = - 0
885      Y,  SUBCD(:CAANLP)
886          $ = MG[10], P
887      Y,  JUMP(2)
888          SE33
889          CPAGEHEADING
890          $ = MG[14]                       " RETURN FROM CHEADS
891          $ - 31, P
892      N,  JUMP(11)
893          MG[14] = $                       " OPVOER := OPVOER - 31
894          $ = MG[21]
895          F = MT[6]
896          MC = $
897          SUBCD(:PSE52)                    " FILL REAL ON INVARIANT ADDRESS
898          G = UBLOCKSTR
899          $ = 2
900          MG[21] + $
901          JUMP(2)
902          1
903          ' 370 000 000'

904          A = MG[21]
905          MG[22] = A
906          A '*' 511                        " LINENUMBER IN A
907          $ = MG[26]
908      U,  A - 433, P
909      Y,  $ - :MG[20], Z                   " NOT ENOUGH ROOM ON 5TH SEGMENT?
910      N,  JUMP(7)
911          $ = 59
912          MC = $
913          SUBC(:MT( - 75))
914          $ = - 1
915          $ + MC[- 1], P
916      Y,  JUMP(- 5)
917          JUMP( - 37)
918          $ = MG[9]                        " :CBUF
919          MG[25] = $                       " INITIALISE RULING BUFFERADDRESS
920          A = 0
921          MS = A                           " INITIALISE FIRST BUFFERWORD
922          MG[11] = - A                     " DUBBEL := EALSE
923          $ = 1                            " ENTRY FOR 0 < POS < 144
924          PLUS$(MG[12])                    " PREINCREASE POS
925          $ '*' 3, Z                       " NEXT BUFFERWORD TO BE FILLED?
926          A = M(B - 2)                     " GET LP REPR, OF SYMBOL FROM STACK
927      N,  JUMP(5)
928          $ = 1
929          PLUS$(MG[25])
```

```
930          RCA(6)
931          MS = A
932          JUMP(7)
933          $ - 3, Z
934      Y,  JUMP(3)
935          $ + 1, Z
936      Y,  LUA(7)
937      N,  LUA(14)
938          S = MG[25]
939          MS + A

940          $ = M[B - 3], Z                 " - DOUBLE CHARACTER
941      Y,  JUMP(- 243)                     " TO RETURN VIA LINK
942          A = MG[11], P
943      Y,  JUMP(21)
944          MG[11] = - A                    " DUBBEL := IRUE
945          A = 4
946          MG[27] = A
947          A = MG[14]
948          RCA(6)
949          MG[23] = A
950          A = MG[21]
951          M[B] = A                        " (PRE) STACK INVARIANT ADDRESS
952          A '*' 511
953      U,  A - 471, P                      " NUMBER OF FREE PLACES < 41?
954      N,  JUMP(10)
955          B + 1
956          F = 0
957          SUBCD(:RSE50)                   " MARK SEGMENT " TO BE CONTINUED "
958          G = UBLOCKSTR
959          A = 1
960          PLUSA(MG[26])
961          LCA(9)
962          A + 1                           " MAKE RULING INVARIANT ADDRESS
963          MG[21] = A
964          MG[22] = A
965          S = MG[27]
966          S - MG[12], P
967      Y,  JUMP(11)
968          S = 1
969          PLUSS(MG[22])
970          MC = S
971          G = MG[23]
972          SUBCD(:RSE50)

973          G = UBLOCKSTR
974          S = 0
975          MG[23] = S
976          S = 4
977          PLUSS(MG[27])
978          JUMP(- 13)
979          S '*' 3, Z
980          A = M[B - 3]
981      Y,  RCA(6)
982      Y,  JUMP(5)
983      U,  S - 1, Z
984      Y,  JUMP(3)
985          S - 3, Z
986      Y,  LUA(14)
987      N,  LUA(7)
```

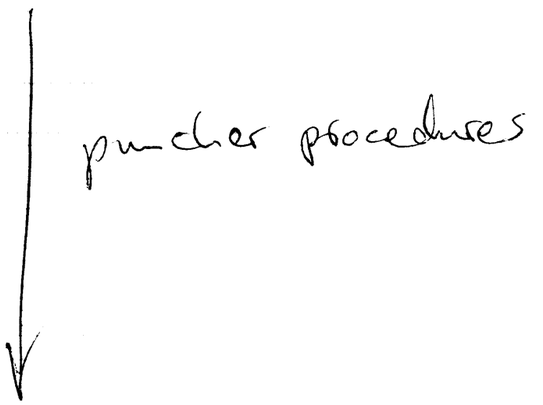
```
 988          MG[23] + A                          " FILL SECONDARY BUFFER
 989          JUMP(- 291)                         " TO RETURN VIA LINK
 990          0                                   " SKIP(3)
 991          0
 992          0

 993 " CREGTRANS(71)                                          → transport line to output segment
 994          SUBCD(:U:MMOBTAL)
 995      W,  A = MS[11], R                       " DUBBEL?
 996      Y,  JUMP(5)
 997          A = MS[14]
 998          RCA(6)
 999          S = MS[9]
1000          M$ + A                              " FILL IN LINEFEED SPECIFICATION
1001          JUMP(14)
1002          A = M$[27]
1003          RUA(2)                              " GRENS ; 4(= NUMBER OF WORDS)
1004          S = M$[21]
1005          G = A
1006          MC = S
1007          SUBCD(:RSE50)                       " FILL IN LENGTH OF SECOND LINE
1008          G = UBLOCKSTR
1009          S = 1
1010          RLUSS(MG[22])
1011          MC = S
1012          S + 1
1013          MG[21] = S
1014          G = MG[23]                          " PLACE LAST DOUBLE PRINTING ON SEGMENT
1015          SUBCD(:RSE50)
1016          G = UBLOCKSTR
1017          S = MG[21]
1018          M[B] = S                            " (PRE) STACK FOR CALL OF $E50
1019          A = MG[25]                          " RULING BAFFER ADDRESS
1020          A - MG[9]                           " - BEGINADDRESS, I.E, NUMBER OF BUFFERWORDS FILLED - 1
1021          S '*' 511
1022          S - 509                             " ROOM NEEDED > ROOM AVAILABLE
1023      U,  A + S, R
1024      N,  JUMP(10)
1025          B + 1                               " ACTUALISE PRESTACK
1026          F = 0
1027          SUBCD(:RSE50)
1028          G = UBLOCKSTR

1029          S = 1
1030          RLUSS(MG[26])
1031          LCS(9)
1032          S + 1
1033          MG[21] = S
1034          JUMP( - 16)
1035          S = MG[26]
1036          SUBCD(:UHOLYRMSEG)
1037          S = MS
1038          S = MS[2]
1039          MC = S                              " STACK CRB
1040          S = MG[21]
1041          S '*' 511
1042          S + MC[- 1]                         " ADD LINENUMBER
1043          A + 1
1044          MS = A                              " FILL IN NUMBER OF WORDS
1045          S + 1                               " INCREASE STORE-ADDRESS
```

```
1046          A = MG[9]                          " INITIALISE FETCH-ADDRESS
1047          F = MA
1048          MS = F
1049          A + 2
1050          S + 2
1051          G = UBLOCKSTR
1052      U,  A - MG[25], F
1053      N,  JUMP(- 7)
1054          A = MG[25]
1055          A - MG[9]
1056          A + 2
1057          MG[21] + A
1058          S = 0
1059          MG[14] = S
1060          MG[12] = - S
1061          S = MG[26]
1062          SUBCD(:URROEANATION)
1063          SUBCD(:UMORTAL)
1064          GOTOR(MC[- 1])

1065  CTRRROC1(1)
1066          CTRRROC1
1067  CRUTEXT2(2)
1068          S = 2
1069          JUMR(1)
1070  CRUTEXT1(45)
1071          S = 1
1072          CGL1 = S
1073          A = 1
1074          S = :MA[3]
1075          SE18
1076          S = 13
1077      U,  S - MA[ - 6], Z
1078      N,  SUBCD(:RSE19)                      " WRONG NO. OF PARAMETERS
1079          S - 13, R                          " TWO PARAMETERS ?
1080      Y,  F = :MA[ - 15]
1081      Y,  SUBCD(:RSE20)                      " CHECK STREAMNUMBER
1082          F = :MA[ - 11]
1083          SUBCD(:RSE29)                      " CHECK FORMAL STRING
1084          S = 25
1085          SE1                                " ANONYMOUS WORKING SPACE
1086              N,  S = CGL1                           " RETRIEVE STREAMNUMBER
1087      N,  JUMP(9)
1088          DOS(MA[ - 15])
1089          DO(ERV)
1090      U,  S = F, Z
1091      N,  SUBCD(:RSE88)
1092          S = G
1093      U,  S -UNUMTRMSTR, R
1094      N,  S - 0, E
1095      Y,  S = 4
1096      Y,  SUBCD(:CALARM)
1097          G = MA[ - 9]
1098          SUBC(:MT[22])                      " EVALUATE :HANDLE
1099      Y,  JUMP(2)
1100          CMEEN
1101          CBEGINTRDOC
1102          SUBCD(:UINSTV)
1103          F = :MC[ - 3]
1104          SUBCD(:UDNSS)
```



```
1105      U,   A - 510, Z                    " END OF STRING ?
1106      N,   MC = A
1107      N,   SUBC(:MT[127])                " PUSYM
1108      N,   JUMP( - 6)
1109           B - 3
1110           S = 100
1111           SUBCD(:UMORTAL)
1112           GBILL + S, P
1113           SE115
1114           DO(MD[ - 1])
1115           SE71
1116 CPUTEXT1(11)
1117           A = 1
1118           S = :MA[3]
1119           SE18
1120           S = 17
1121           JUMP( - 44)                   " TO MERGE WITH PUTEXT1
1122           $ +UTRSTREAMS                 " EVALUATE :HANDLE
1123           S = MS[ - 1]
1124           UBLOCKSTR = S
1125           SUBCD(:UMMORTAL)
1126      U,   A = MS[6], P
1127           GOTO(MC[ - 1])

1128 CONRUNUM(28)        → continue punching of number
1129           S = :MG[4]
1130           A = :MC[ - 144]
1131           G = UBLOCKSTR
1132           A + MG[19]
1133           A - S, P                      " SPACE ON LINE INSUFFICIENT ?
1134           MC = S                        " POINTER
1135      N,   JUMP(4)
1136           S = 119
1137           MC = S
1138           SUBC(:MT[99])                 " PUSYM + 1
1139           S = M[B - 1]
1140           A = MS
1141      U,   A - 9, P                      " - DIGIT ?
1142      Y,   A + :MT[ - 1]
1143      Y,   A = MA
1144           MC = A
1145           SUBC(:MT[92])
1146           S = 1
1147           PLUSS(M[B - 1])
1148      U,   S - :MC[ - 2], P
1149      N,   JUMP( - 10)
1150           S = 100
1151           JUMP( - 39)                   " TO INCREASE BILL
1152           64                            " +
1153           65                            " -
1154           88                            " .
1155           89                            " 10
1156           93                            " BLANK SPACE

1157 CRUSPACE1(2)
1158           SUBC(:MT[184])
1159           JUMP(4)
1160 CRUSPACE2(2)
1161           S = 2
1162           JUMP(1)
```

```
1163 CRUSPACE1(16)
1164          $ = 1
1165          SUBC(:MT[136])
1166          F = 0, P                          " N > 0 ?
1167          $ = 5
1168      N,  JUMP( - 53)                       " TO INCREASE BILL
1169          MC = $
1170          MC = G
1171          $ = 93
1172          MC = $
1173          SUBC(:MT[66])
1174          F = - 1
1175          G + MC[ - 1], P
1176          $ = 5
1177          $ + MC[ - 1]
1178      Y,  JUMP( - 10)
1179          JUMP( - 64)                       " TO INCREASE BILL

1180 CRUSYM1(2)
1181          SUBC(:MT[164])
1182          JUMP(4)
1183 CRUSYM2(2)
1184          $ = 2
1185          JUMP(1)
1186 CRUSYM1(9)
1187          $ = 1
1188          SUBC(:MT[116])
1189          $ = 0
1190          MC = - $                          " BOOLEAN VALUE FALSE
1191          MC = G, Z
1192      Y,  M[B - 1] = $                      " TO MAKE ONE REPRESENTATION FOR 0
1193          SUBC(:MT[49])
1194          G = MC[ - 1]
1195          JUMP(45)                          " TO CALCULATE SAME PRICE AS FOR CRUTAB
1196 CRUCHARE1(2)
1197          SUBC(:MT[151])
1198          JUMP(4)
1199 CRUCHARE2(2)
1200          $ = 2
1201          JUMP(1)
1202 CRUCHARE1(9)
1203          $ = 1
1204          SUBC(:MT[103])
1205          $ = 0
1206          MC = - $                          " MERGE FROM RUNLCR
1207          MC = G, Z
1208      Y,  M[B - 1] = $
1209          SUBC(:MT[187])
1210          G = MC[ - 1]
1211          JUMP(32)                          " TO CALCULATE SAME PRICE AS FOR CRUTAB

1212 CRUNLCR1(2)
1213          SUBC(:MT[145])
1214          JUMP(4)
1215 CRUNLCR2(2)
1216          $ = 2
1217          JUMP(1)
1218 CRUNLCR1(4)
1219          $ = 1
1220          SUBC(:MT[164])
```

```
1221            F = 119                         " VALUE FOR NLCR
1222            JUMP( - 14)                     " TO MERGE WITH CRUCHARF1
1223 CRUTAB1(2)                                 " TO MERGE WITH CRURUNOUT
1224            SUBC(:MT[137])
1225            JUMP(4)
1226 CRUTAB2(2)
1227            $ = 2
1228            JUMP(1)
1229 CRUTAB1(173)
1230            $ = 1
1231            SUBC(:MT[156])
1232            G = UBLOCKSTR
1233            $ = MG[18]
1234       U,   $ - 2, Z
1235       N,   A = 93
1236       N,   MC = A
1237       N,   SUBC(:MT[18])
1238            $ = MG[19]
1239            $ + 9
1240            $ '*' - 7
1241       U,   $ - 144, F
1242       Y,   A = 26
1243       Y,   SUBCD(:UTREILL[1])
1244       Y,   $ = 8
1245       Y,   MG[19] = $
1246       Y,   MG[20] = $
1247       N,   $ - MG[19]
1248       N,   MG[19] + $
1249       N,   MG[20] + $
1250            $ = 20
1251            JUMP( - 124)                    " TO INCREASE BILL

1252  " BUSYM(62)                               " 62 + 43 + 7 + 24 + 15
1253            G = UBLOCKSTR
1254            $ = MG[18]
1255       U,   $ - 2, Z
1256       Y,   JUMP(11)
1257            $ + 0, Z
1258       N,   $ = 5
1259       N,   SUBCD(:CALARM)                  " TELEXCODE NOT IMPLEMENTED
1260            MG[19] = $
1261            MG[20] = $
1262            $ = 1
1263            MG[21] = $
1264            A = 122
1265            SUBCD(:UTREILL[1])
1266            $ = 2
1267            MG[18] = $
1268            $ = M[B - 2]
1269       U,   $ - 93, Z
1270       Y,   JUMP(152)                       " TO MERGE WITH CHARF1
1271            $ - 119, Z
1272       Y,   JUMP(156)                       " TO MERGE WITH CHARF2
1273            $ = M[B - 2], F
1274       U,   $ - 130, E
1275       Y,   JUMP(171)                       " TO MERGE WITH CHARF3
1276            RCS(1), F
1277       N,   $ '*' 511
1278            $ + :MT[225]
1279            $ = MS
```

this is actually a line of comment

```
1280      Y,  RUS(12)
1281      U,  $ '*' '1600', Z
1282      Y,  JUMP(167)                          " TO MERGE WITH CHARE4
1283      U,  $ '*' 512, Z
1284          $ '*' 511
1285      Y,  M[B - 2] = $
1286      Y,  JUMP(152)                          " TO MERGE WITH CHARE5
1287          $ + :CTABWDL
1288          A = MG[19]
1289      U,  A - 134, P                         " - ROOM FOR WORDDELIMITER?
1290          MC = $                             " SAVE LISTPOINTER
1291      N,  JUMP(6)
1292          A = 26
1293          SUBCD(:UTREILL[1])
1294          $ = 0
1295          MG[19] = $
1296          MG[20] = $
1297          $ = M[B - 1]
1298          A = MS
1299          $ = 31
1300          $ '*' A, Z                         " COLLATE OUT ONE LETTER, DONE?
1301      Y,  JUMP(11)
1302          RUA(5)
1303          M[B - 3] = A
1304          $ + 137
1305          MC = $
1306          SUBC(:MT[132])                     " CALL CHARE5
1307          A = M[B - 3], Z
1308      N,  JUMP( - 10)
1309          A + 0, P
1310      N,  $ = M[B - 1]
1311      N,  A = MS[1]
1312      N,  JUMP( - 14)
1313          B - 2
1314          GOTOB(MC)

1315  " CRUKORN(43)
1316          CGL1 = $
1317          $ = MC[ - 1]
1318          CGL2 = $                           " SAVE LINK
1319          A = 1
1320          $ = :MA[3]
1321          $E18
1322          $ = 13
1323      U,  $ - MA[ - 6], Z
1324      N,  SUBCD(:PSE19)
1325      U,  $ - 13, P
1326      Y,  F = :MA[ - 15]
1327      Y,  SUBCD(:PSE20)
1328          F = :MA[ - 11]
1329          SUBCD(:PSE20)
1330          $ = 25
1331          $E1
1332          $ = CGL2
1333          MC = $
1334      N,  $ = CGL1
1335      N,  JUMP(9)
1336          DOS(MA[ - 15])
1337          DO(ERV)
1338      U,  $ = F, Z
```

```
1339      N,  SUBCD(:@SE88)
1340          $ = G
1341      U,  $ - UNUMTRRMSTR, R
1342      N,  $ - 0, E
1343      Y,  $ = 4
1344      Y,  SUBCD(:CALARM)
1345          MC = $                          " SAVE STREAMNUMBER
1346          DOS(MA[ - 11])
1347          DO(ERV)
1348      U,  $ = F, Z
1349      N,  SUBCD(:@SE88)                   " PARAMETER N IN G
1350          $ = MC[ - 1]
1351          SUBC(:MT[ - 212])
1352      Y,  JUMP(2)
1353          CHEEN
1354          CBEGINTRDOC
1355          $ = MC[ - 1]                    " RETURN VIA LINK
1356          $ '*' 511                       " WITH PARAMETER IN G
1357          $ - 272
1358          JUMP($)
1359  " CRUKORIN(7)
1360          $ = MC[ - 1]
1361          CGL2 = $
1362          A = 1
1363          $ = :MA[3]
1364          SE18
1365          $ = 17
1366          JUMP( - 43)                     " TO MERGE WIT CRUKORN
1367  " CRUKORI(24)
1368          $ = MC[ - 1]
1369          CGL2 = $
1370          A = 1
1371          $ = :MA[3]
1372          SE18
1373          $ = 13
1374      U,  $ - MA[ - 6], Z
1375      N,  SUBCD(:@SE19)
1376          F = :MA[ - 11]
1377          SUBCD(:@SE20)
1378          $ = 25
1379          SE1
1380          $ = CGL2
1381          MC = $
1382          DOS(MA[ - 11])
1383          DO(ERV)
1384      U,  $ = F, Z
1385      N,  SUBCD(:@SE88)
1386          $ = G
1387      U,  $ - UNUMTRRMSTR, R
1388      N,  $ - 0, E
1389      Y,  $ = 4
1390      Y,  SUBCD(:CALARM)
1391          JUMP( - 39)                     " TO MERGE WITH CRUKORN
1392  " CRUKOR(15)
1393          CGL1 = $                        " SAVE STREAMNUMBER
1394          $ = MC[ - 1]                    " LINK
1395          CGL2 = $
1396          A = 1
1397          $ = :MA[3]
```

```
1398          $E18
1399          $ = MA[ - 6]
1400      U,  $ - 9, Z
1401      N,  $UBCD(:#SE19)
1402          $ = 25
1403          $E1
1404          $ = CGL2
1405          MC = $
1406          $ = CGL1
1407          JUMP( - 54)                    " TO MERGE WITH CRUKORN

1408 PUCHARE(166)
1409          G = UBLOCKSTR
1410          $ = MG[18]
1411      U,  $ - 2, Z                       " PUTYPE = 2
1412      Y,  JUMP(12)
1413      U,  $ - 3, Z                       " TELEXCODE RULING?
1414      Y,  A = 0                          " PRELIMINARY VALUE
1415      Y,  $UBCD(:UTREILL[1])
1416          $ = 2
1417          MG[18] = $
1418          A = 122
1419          $UBCD(:UTREILL[1])
1420          $ = 1
1421          MG[21] = $                     " BOOLEAN LOWERCASE
1422          $ = 0
1423          MG[19] = $                     " POSITION ON LINE
1424          MG[20] = $                     " SPACES IN STORAGE
1425          $ = M[B - 2]
1426      U,  $ - 93, Z
1427      N,  JUMP(11)
1428          $ = 1                          " MERGE1
1429          MG[20] + $
1430          PLUS$(MG[19])
1431          $ - 144, P
1432      N,  B - 1
1433      N,  GOTOR(MC)
1434          MG[19] = $                     " MERGE2
1435          MG[20] = $
1436          A = 26, Z
1437          SUBCD(:UTREILL[1])
1438          JUMP( - 7)
1439          $ - 119, Z
1440      Y,  JUMP( - 7)
1441          $ = M[B - 2], P
1442      U,  $ - 511, E                     " ¬ VALUE ALLOWED?
1443      Y,  JUMP(8)                        " TO TREAT WRONG VALUE
1444          $ '*' 127                      " MERGE 5
1445          RCS(1), P
1446      N,  $ '*' 511
1447          $ + :MT[61]
1448          $ = M$
1449      Y,  RUS(12)
1450      U,  $ '*' '1600', Z
1451      Y,  JUMP(3)
1452          $ = 0                          " MERGE 3
1453          M[B - 3] = $                   " PUSYM := TRUE
1454          JUMP(52)                       " TO B - 1
1455          $ '*' '3777'                   " MERGE 4
1456          MC = $
```

```
1457          A = MG[19]
1458          A - 144, Z
1459      N,  JUMP(4)
1460          MG[19] = A
1461          MG[20] = A
1462          A = 26
1463          SUBCD(:UTREILL[1])
1464      N,  A = MG[20], Z
1465      Y,  JUMP(14)
1466          JUMP(8)                    " IN CASE TABS ARE WANTED, REPLACE BY: $ = MG[19]
1467          $ - A
1468          $ + 9
1469          $ '*' - 7
1470          $ - MG[19], P
1471      N,  MG[20] = - $
1472      N,  A = 62
1473      N,  SUBCD(:UTREILL[1])
1474      N,  JUMP( - 11)
1475          A = 16
1476          SUBCD(:UTREILL[1])
1477          A = - 1
1478          MG[20] + A, P
1479      Y,  JUMP( - 5)
1480          A = 1
1481          MG[19] + A
1482          $ = M[B - 3]               " TAKE ORIGINAL CHARE VALUE
1483      U,  $ - 127, P                 " UNDERLINING/STROKE?
1484      N,  JUMP(13)
1485      U,  $ - 383, P                 " BOTH
1486      Y,  A = 14
1487      Y,  SUBCD(:UTREILL[1])
1488      Y,  JUMP(3)
1489      U,  $ - 255, P
1490          $ = MG[21], E
1491      N,  JUMP(4)
1492          A = MG[21]                 " CHANGE CASE
1493          MG[21] = - A               " + OR - 1
1494          A + 123
1495          SUBCD(:UTREILL[1])
1496          A = 14
1497          SUBCD(:UTREILL[1])
1498          $ = M[B - 1]
1499          $ - '1777', P
1500          A = MG[21], E
1501      Y,  MG[21] = - A
1502      Y,  A + 123
1503      Y,  SUBCD(:UTREILL[1])
1504          A = MC[ - 1]
1505          A '*' 255
1506          SUBCD(:UTREILL[1])
1507          B - 1
1508          GOTOR(MC)

1509  " CFLEXTAB(66)
1510          '0040 0001'                " 0     1
1511          '0002 0023'                " 2     3
1512          '0004 0025'                " 4     5
1513          '0026 0007'                " 6     7
1514          '0010 0031'                " 8     9
1515          '0141 0142'                " A     B
```

conversion table

```
1516        '0163 0144'                    " C     D
1517        '0165 0166'                    " E     F
1518        '0147 0150'                    " G     H
1519        '0171 0121'                    " I     J
1520        '0122 0103'                    " K     L
1521        '0124 0105'                    " M     N
1522        '0106 0127'                    " O     P
1523        '0130 0111'                    " Q     R
1524        '0062 0043'                    " S     T
1525        '0064 0045'                    " U     V
1526        '0046 0067'                    " W     X
1527        '0070 0051'                    " Y     Z
1528        '0340 2141'                    "       Ä
1529        '2142 2163'                    " Ö     Ç
1530        '2144 2165'                    " Ü     È
1531        '2166 2147'                    " É     Ğ
1532        '2150 2171'                    " Ħ     İ
1533        '2121 2122'                    " J     K
1534        '2103 2124'                    " L     M
1535        '2105 2106'                    " N     Ø
1536        '2127 2130'                    " Þ     Q
1537        '2111 2062'                    " R     Ş
1538        '2043 2064'                    " T     U
1539        '2045 2046'                    " V     W
1540        '2067 2070'                    " X     Y
1541        '2051 0340'                    " Z
1542        '0160 0100'                    " +     -
1543        '2002 2023'                    " *     /
1544        '0332 0520'                    " ↑     ↓
1545        '2004 0506'                    " =     ≠
1546        '0061 0310'                    " <     ≤
1547        '2061 0312'                    " >     ≥
1548        '2100 0306'                    " ¬     ≡
1549        '0314 2001'                    " ⊐     ∨
1550        '2040 1000'                    " ^     GOTO
1551        '1001 1002'                    " EOR   STEP
1552        '1003 1004'                    " UNTIL WHILE
1553        '1005 0133'                    " DO    ,
1554        '0153 0073'                    " .     ␣
1555        '2153 2025'                    " :     ;
1556        '1006 0020'                    " COMPLEX SPATIE
1557        '1010 1011'                    " IF    THEN
1558        '1012 1013'                    " ELSE  COMMENT
1559        '2010 2031'                    " (     )
1560        '2026 2007'                    " [     ]
1561        '0510 0512'                    " ⌊     ⌋
1562        '1015 1016'                    " BEGIN END
1563        '1017 1020'                    " OWN   REAL
1564        '1021 1023'                    " INTEGER BOOLEAN
1565        '1025 1027'                    " STRING ARRAY
1566        '1030 1032'                    " PROCEDURE SWITCH
1567        '1034 1035'                    " LABEL VALUE
1568        '1036 1037'                    " TRUE  FALSE
1569        '1040 0032'                    " PROGEND NLCR
1570        '2073 2160'                    " '     "
1571        '2133 1045'                    " ?     OMGEKEERD
1572        '1042 1046'                    " LIBRARY INVSLASH
1573        '0335 0535'                    " _     |
1574        '1044 0327'                    " NLCR ▲
1575        '1050 0000'                    " ALGOL
```

```
1576 CRUNOUT(11)
1577          S = 1
1578          SUBC(:MT[ - 194])
1579          S = 80
1580          MC = S
1581          A = 0
1582          SUBCD(:UTRFILL)
1583          S = - 1
1584          S + MC[ - 1], R
1585     Y,   JUMP( - 6)
1586          S = 300
1587          JUMP( - 463)                    " TO INCREASE BILL

1588 CTRPROC2(55)
1589          CTRPROC2
1590          CGL1 = S                        " STORE STREAMNUMBER, CRUKORN(47)
1591          S = MC[ - 1]
1592          CGL2 = S                        " SAVE LINK
1593          A = 1
1594          S = :MA[3]
1595          SE18
1596          S = 13
1597     U,   S - MA[ - 6], Z                 " CHECK NUMBER OF PARAMETERS, MERGING POINT
1598     N,   SUBCD(:RSE19)
1599     U,   S - 13, R                       " TWO PARAMETERS ?
1600     Y,   F = :MA[ - 15]
1601     Y,   SUBCD(:RSE20)
1602          F = :MA[ - 11]
1603          SUBCD(:RSE20)
1604          S = 25
1605          SE1
1606          S = CGL2
1607          MC = S
1608     N,   S = CGL1
1609     N,   JUMP(9)
1610          DOS(MA[ - 15])                  " EVALUATE STREAMNUMBER
1611          DO(ERV)
1612     U,   S = F, Z
1613     N,   SUBCD(:RSE88)
1614          S = G
1615     U,   S - UNUMTRRMSTR, R
1616     N,   S - 0, E
1617     Y,   S = 4
1618     Y,   SUBCD(:CALARM)
1619          MC = S                          " SAVE STREAMNUMBER
1620          DOS(MA[ - 11])                  " EVALUATE N
1621          DO(ERV)
1622     U,   S = F, Z
1623     N,   SUBCD(:RSE88)                   " PARAMETER N IN G
1624          S = MC[ - 1]
1625          S + UTRSTREAMS
1626          S = MS[ - 1]
1627          UBLOCKSTR = S
1628          SUBCD(:UIMMORTAL)
1629     U,   A = MS[6], R                    " DOCUMENT UNDER CONSTRUCTION ?
1630     Y,   JUMP(2)
1631          CHEEN
1632          CBEGINTRDOC
1633          S = MC[ - 1]                    " RETURN VIA LINK
```

by a convention every first word of a program segment contains its own invariant address

more punching procedures

```
1634          S '*' 511                          " WITH PARAMETER IN G
1635          S - 48
1636          JUMP(S)
1637 "CRUKORIN(7)
1638          S = MC[ - 1]
1639          CGL2 = S
1640          A = 1
1641          S = :MA[3]
1642          SE18
1643          S = 17                             " INDICATE 2 PARAMETERS
1644          JUMP( - 47)                        " TO MERGE WITH CRUKORN

1645 CRUOCT1(2)
1646          SUBC(:MT[ - 8])
1647          JUMP(4)
1648 CRUOCT2(2)
1649          S = 2
1650          JUMP(1)
1651 CRUOCT1(23)
1652          S = 1
1653          SUBC(:MT[ - 60])
1654          A = G, Z                           " COPY PARAMETER IN A
1655      Y,  A = 0
1656          G = UBLOCKSTR
1657          S = MG[18], Z                      " PUTYPE = 0
1658      Y,  JUMP(8)
1659          MC = A                             " SAVE OCTAD
1660      U,  S - 2, Z                           " FLEXOCODE RULING ?
1661      Y,  A = 11                             " STOPCODE
1662      N,  A = 0                              " PRELIMINARY VALUE TO END TELEXCODE
1663          SUBCD(:UTREILL[1])
1664          S = 0
1665          MG[18] = S
1666          A = MC[ - 1]
1667          A '*' 255
1668          SUBCD(:UTREILL[1])
1669          S = 15
1670          SUBCD(:UMORTAL)
1671          GBILL + S, R
1672          SE115
1673          DO(MD[ - 1])
1674          SE71
1675 CRURUNOUT1(2)
1676          SUBC(:MT[ - 35])
1677          JUMP(4)
1678 CRURUNOUT2(2)
1679          S = 2
1680          JUMP(1)
1681 CRURUNOUT1(17)
1682          S = 1
1683          SUBC(:MT[ - 87])
1684          F - 2048, R
1685      Y,  F = 2048
1686      N,  F+ 2048, R
1687          S = 3
1688      N,  JUMP( - 16)
1689          MC = S
1690          MC = G
1691          A = 0
1692          SUBCD(:UTREILL)
```

```
1693          F = - 1
1694          G + MC[ - 1], R
1695          $ = 3
1696          $ + MC[ - 1]
1697      Y,  JUMP( - 9)
1698          JUMP( - 26)                          " TO INCREASE GBILL

1699  CENDTRDOC2(2)
1700          $ = 2
1701          JUMP(1)
1702  CENDTRDOC1(7)
1703          $ = 1
1704          CGL1 = $
1705          A = 1
1706          $ = :MA[3]
1707          $E18
1708          $ = 9
1709          JUMP(4)                              " TO MERGE WITH CENDTRDOC1
1710  CENDTRDOC1(46)      end of punches document
1711          A = 1
1712          $ = :MA[3]
1713          $E18
1714          $ = 13
1715      U,  $ - MA[ - 6],   Z
1716      N,  SUBCD(:RSE19)
1717      U,  $ - 9, R
1718      Y,  F = :MA[ - 11]
1719      Y,  SUBCD(:RSE20)
1720          $ = 25
1721          $E1
1722      N,  $ = CGL1
1723      N,  JUMP(9)
1724          DOS(MA[ - 11])
1725          DO(ERV)
1726      U,  $ = F, Z
1727      N,  SUBCD(:RSE88)
1728          $ = G
1729      U,  $ - UNUMTRMSTR, R
1730      N,  $ - 0, E
1731      Y,  $ = 4
1732      Y,  SUBCD(:CALARM)
1733          $ + UTRSTREAMS
1734          G = MS[ - 1]
1735          UBLOCKSTR = G
1736          SUBCD(:UIMMORTAL)
1737      U,  A = MG[6], R
1738      N,  JUMP(16)
1739          $ = MG[18], Z                        " PUTYPE = 0 ?
1740      Y,  JUMP(4)
1741      U,  $ - 2, Z
1742      Y,  A = 11
1743      N,  A = 0
1744          SUBCD(:UTRFILL[1])
1745          A = MG[17]                           " BUILDING UP WORD
1746      U,  A '*' UCON1, Z                       " SUPPLY WITH END MARKERS
1747      N,  LCA(9)
1748      N,  A '+' 1, R
1749      N,  JUMP( - 4)
1750          MG[17] = A                           " STORE BUILDING UP WORD
1751          SUBCD(:UFILLSEGMENT)                 " FILL WORD ON SEGMENT
```

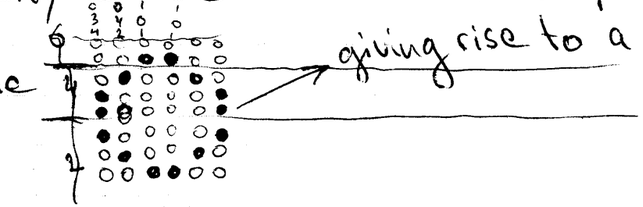
```
1752          $UBCD(:USH4)
1753          $UBCD(:USH8)
1754          $UBCD(:UTREXVWITADRES)
1755          $ = 100
1756          JUMP( - 81)                          " TO EXIT
1757 CBEGINTRDOC(101)
1758          MC = G
1759          UTREXPROOD
1760          A = USPECIALRM, R
1761      Y,  A = 1
1762      N,  A = ULOANRUN
1763      N,  A + UCLAIMRUN
1764          A - UDOCRUN, R
1765      N,  $UBCD(:UTREXVWITADRES)
1766      N,  $UBCD(:UMORTAL)
1767      N,  $ = 6
1768      N,  $UBCD(:CALARM)
1769          A = 1
1770          UDOCRUN + A
1771          MS[6] = A
1772          $ = USPECIALRM, R
1773      Y,  $ + 1
1774      Y,  UAMVAR = $
1775      Y,  $UBCD(:UPUPM)
1776          UTREXVWIT
1777      Y,  $UBCD(:UPOWNSEMADRES)
1778          $ = UBLOCKSTR
1779          A = 0
1780          MS[18] = A
1781          $ = 3000
1782          GBILL + $, R
1783          SE115
1784          $ = URUNNUM
1785          A = 8
1786          CHEEN
1787          CRUVISNUM
1788          SUBC(:MT[37])
1789          $ = UDATE
1790          A = 8
1791          CHEEN
1792          CRUVISNUM
1793          SUBC(:MT[32])
1794          G = UPROGID
1795          $UBCD(:UINSTV)
1796          F = :MC[ - 3]
1797          $UBCD(:UDNSS)
1798      U,  A - 510, Z
1799      N,  SUBC(:MT[43])
1800      N,  JUMP( - 5)
1801          B - 3
1802          $ = UBLOCKSTR
1803          A = MS[9]                            " DOCUMENT NUMBER
1804          MC = A                               " AND SAVE ON STACK
1805          $ = MS[5]
1806          LCS(9)
1807          $ '*' 511                            " COLLATE OUT STREAMNUMBER
1808          A = 1
1809          CHEEN
1810          CRUVISNUM
1811          SUBC(:MT[14])
```

```
1812          S = MC[ - 1]
1813          A = 3
1814          CHEEN
1815          CRUVISNUM
1816          SUBC(:MT[9])
1817          S = 150
1818          MC = S
1819          A = 0
1820          SUBCD(:UTREILL)
1821          S = - 1
1822          S + MC[ - 1], Z
1823      N,  JUMP( - 6)
1824          G = MC[ - 1]
1825          CTERUG

1826          S = MC[ - 1]                    " PUNCH LEGIBLE NUMBER
1827          MG[ - 8] = S                    " TRANSFER LINK TO BOTTOM LEVEL
1828          S = :MG[4]                      " ADDRESS OF FIRST SYMBOL
1829          MC = G                          " SAVE G
1830          MC = S                          " SAVE POINTER
1831          A = MS                          " GET SYMBOL
1832      U,  A - 9, P                        " TRANSFORM NON-DIGIT
1833      Y,  A = 93                          " INTO SPACE
1834          SUBC(:MT[8])                    " AND PUNCH LEGIBLE CHARACTER
1835          S = 1
1836          S + MC[ - 1]
1837      U,  S - :MC[ - 2], P
1838      N,  JUMP( - 9)
1839          B = MC[ - 1]
1840          B - 7
1841          GOTOR(MC[ - 1])
1842          CVISRU
1843          G = MT[ - 2]                    " ENTRYPOINT
1844      U,  A - 62, P                       " ¬ DIGIT OR LETTER ?
1845      Y,  A = 36                          " VALUE FOR BLANK SPACE
1846      Y,  JUMP(2)
1847      U,  A - 36, E                       " ¬ CAPITAL LETTER ?
1848      N,  A - 27                          " ALL LETTERS IN ONE TYPE
1849          LUA(2)                          " MULTIPLICATION FOR UNIFORM STRINGLENGTH 4
1850          G + A                           " SELECT STRING REPRESENTING THIS CHARACTER
1851          SUBCD(:UINSTV)
1852          P = :MC[ - 3]
1853          SUBCD(:UDNSS)
1854      U,  A - 510, Z
1855      N,  SUBCD(:UTREILL)
1856      N,  JUMP( - 5)
1857          B - 3
1858          GOTOR(MC[ - 1])

1859  CVISRU(147)                             " STRING LAYOUT LEESBARE PONSINGEN
1860          '034'                           " 0
1861          '042 101 101'
1862          '042 034 000'
1863          '000 776 000'
1864          '040'                           " 1
1865          '177 000 000'
1866          '776 000 000'
1867          0
1868          '043'                           " 2
1869          '105 111 061'
```

table of directly readable characters to be punched heading a tape document



```
1870        '776'
1871        0
1872        '141'                           " 3
1873        '101 115 177'
1874        '776'
1875        0
1876        '174'                           " 4
1877        '004 004 037'
1878        '776'
1879        0
1880        '002'                           " 5
1881        '161 121 121'
1882        '116 000 000'
1883        '776 000 000'
1884        '036'                           " 6
1885        '051 111 106'
1886        '776'
1887        0
1888        '100'                           " 7
1889        '100 117 160'
1890        '776'
1891        0
1892        '067'                           " 8
1893        '115 115 067'
1894        '776'
1895        0
1896        '060'                           " 9
1897        '111 111 076'
1898        '776'
1899        0
1900        '077'                           " A
1901        '10 110 110'
1902        '077 000 000'
1903        '776 000 000'
1904        '177'                           " B
1905        '111 111 111'
1906        '167 000 000'
1907        '776 000 000'
1908        '034'                           " C
1909        '042 101 101'
1910        '101 000 000'
1911        '776 000 000'
1912        '177'                           " D
1913        '101 101 101'
1914        '042 034 000'
1915        '000 776 000'
1916        '177'                           " E
1917        '111 111 101'
1918        '776'
1919        0
1920        '177'                           " F
1921        '110 110 100'
1922        '776'
1923        0
1924        '036'                           " G
1925        '041 105 105'
1926        '107 000 000'
1927        '776 000 000'
1928        '177'                           " H
1929        '010 010 010'
```

```
1930          '010 177 000'
1931          '000 776 000'

1932          '000 000 101'                 " I
1933          '177 101 000'
1934          '000 776 000'
1935          0
1936          '001'                         " J
1937          '001 176 000'
1938          '000 776 000'
1939          0
1940          '177'                         " K
1941          '010 030 044'
1942          '102 001 000'
1943          '000 776 000'
1944          '177'                         " L
1945          '001 001 001'
1946          '000 000 776'
1947          0
1948          '177'                         " M
1949          '040 020 020'
1950          '040 177 000'
1951          '000 776 000'
1952          '177'                         " N
1953          '040 020 010'
1954          '004 002 177'
1955          '776'
1956          '076'                         " O
1957          '101 101 101'
1958          '076 000 000'
1959          '776 000 000'
1960          '177'                         " P
1961          '110 110 060'
1962          '000 000 776'
1963          0
1964          '076'                         " Q
1965          '141 121 111'
1966          '076 000 000'
1967          '776 000 000'
1968          '177'                         " R
1969          '110 114 063'
1970          '776'
1971          0
1972          '061'                         " S
1973          '111 111 106'
1974          '776'
1975          0
1976          '100'                         " T
1977          '100 100 177'
1978          '100 100 100'
1979          '776'
1980          '176'                         " U
1981          '001 001 001'
1982          '001 176 000'
1983          '000 776 000'
1984          '170'                         " V
1985          '004 002 001'
1986          '002 004 170'
1987          '000 000 776'
1988          '000 174 002'                 " W
```

```
1989          '001 002 004'
1990          '002 001 002'
1991          '174 000 776'
1992          '000 101 042'                    " X
1993          '024 010 010'
1994          '024 042 101'
1995          '000 776 000'
1996          '100'                            " Y
1997          '040 020 017'
1998          '020 040 100'
1999          '000 000 776'
2000          '103'                            " Z
2001          '105 101 111'
2002          '121 141 000'
2003          '000 776 000'
2004          0
2005          0
2006          '776'                            " BLANK SPACE FOR ANY OTHER CHARACTER
```

```
 0 UJOB1(159)
 1          B = MA[0]
 2          CHEEN
 3          UTELEO
 4          GL18 = G
 5          CHEEN
 6          USRCR                    " SKIP SPACE AND CARRIAGE RETURN
 7      U,  S - 48, Z                " L?
 8      N,  ULIST = - B              " LIST := EALSE
 9      N,  JUMP(3)
10          ULIST = B                " LIST := IRUE
11          CHEEN
12          USRCR                    " SKIP SPACE AND CARRIAGE RETURN
13      U,  S - 130, Z               " ALGOL?
14      Y,  JUMP(2)
15          SE33
16          UJOB2                    " MAYBE THE SECOND JOB?
17          F = - 0                  " INITIALISATION
18          GL19 = G
19          MC[0] = F
20          MC[0] = F
21          F = - 1
22          UKIND = - G              " INDICATION ALGOL
23          G + UWHO
24          MC[0] = F                " STACK UINVWHO[- 1]
25          CHEEN
26          USR
27      U,  S '*' - 63, Z            " LETTER OR DIGIT?
28      N,  F - 93, Z                " SPACE?
29      N,  JUMP(28)
30          A = 1
31          PLUSA(M[B - 2])          " INCREASE COUNTER
32      U,  A - 8, P
33      Y,  JUMP(2)
34      U,  S - 9, P
35      Y,  JUMP(51)                         " TO UNINT
36          A - UNAMELENGTH, P       " IDENTIFICATION STRING ALREADY COMPLETED?
37      Y,  JUMP(16)
38      U,  S - 93, Z                " SPACE?
39      Y,  A = 1
40      Y,  GL19 + A                 " INCREASE SPACE COUNTER
41      Y,  JUMP(12)
42          GL20 = S
43          A = GL19, Z              " ALL PREVIOUS SPACES FILLED?
44      Y,  JUMP(6)
45          S = 93
46          CHEEN
47          UFILLSTRING              " FILL SPACES
48          A = 1
49          GL19 - A
50          JUMP(-8)
51          S = GL20
52          CHEEN                    " FILL SYMBOL IN STRING
53          UFILLSTRING
54          DO(MD[- 1])
55          CHEEN
56          UNEXTSYM                 " NEXT SYMBOL
57          JUMP( - 31)
58      U,  S - 87, Z                        " KOMMA?
59      N,  JUMP(25)
```

this is where anille PM starts to inspect an input tape to see whether it is correctly headed and what kind of a job it is.

```
60          CHEEN
61          USR
62      U,  $ - 9, P
63      Y,  JUMP(23)
64          M[B - 6] + $                    " SET PUCLAIMPUN(= 0 PREFERENCE)
65          CHEEN
66          USR
67      U,  $ - 87, Z
68      N,  JUMP(16)                        " KOMMA?
69          CHEEN
70          USR
71      U,  $ - 9, P
72      Y,  JUMP(14)
73          M[B - 5] + $                    " SET PRECLAIMPLO(= 0 PREFERENCE)
74          CHEEN
75          USR
76      U,  $ - 87, Z
77      N,  JUMP(7)                         " KOMMA?
78          CHEEN
79          USR
80      U,  $ - 9, P
81      Y,  JUMP(5)
82          M[B - 4] + $                    " SET PRECLAIMMAG(= 0 PREFERENCE)
83          CHEEN
84          UTELE1
85      U,  $ - 119, Z              " CARRIAGE RETURN?
86      Y,  JUMP(2)
87          SE33
88          UNINT  ---> wrong heading
89          CHEEN
90          USRCR                   " SKIP SPACE AND CARRIAGE RETURN
91      U,  $ - 120, Z              " APOS?
92      N,  JUMP(6)
93          CHEEN
94          UNEXTSYM                " NEXT SYMBOL
95      U,  $ - 120, Z              " APOS?
96      N,  JUMP(- 4)
97          CHEEN
98          USRCR                   " SKIP SPACE AND CARRIAGE RETURN
99          G = UTRSTREAMS
100         G = MG[0]
101         MG[14] = $              " BACK SYM
102         $ = 93
103         CHEEN
104         UFILLSTRING
105         $ = 510
106         CHEEN
107         UFILLSTRING

108         UDEAF
109         SUBCD(:UCLAIMCONRED)
110         UTREXP
111         A = M[B - 6]
112     U,  A - UNUMTRMSTR, P               " PRECLAIMPUN > NUMBER OF PUNCH STREAMS?
113     N,  $ = M[B - 5]
114     N,  $ - UNUMPLRMSTR, P              " PRECLAIMPLO > NUMBER OF PLOTTERSTREAMS?
115     Y,  JUMP(14)
116         $ = USPECIALRM, P
117     N,  JUMP(2)
118     U,  A - 1, P                        " PRECLAIMPUN > 1?
```

```
119          JUMP(1)
120      U,  A - URUNACCERT, P                " PRECLAIMPUN > PUNACCEPT?
121      N,  S = M[B - 5]
122      N,  $ - UPLOACCEPT, P                " PRECLAIMPLO > PLOACCEPT?
123      Y,  JUMP(6)
124          S = M[B - 4], Z                  " PRECLAIMMAG = 0?
125      Y,  JUMP(7)
126      U,  $ - JMTEREE, P
127      N,  JMTEREE - S
128      N,  JMTCLAIMED = $
129      N,  JUMP(3)
130          UTREXV
131          SE33
132          UNAC
133          S = GL27, Z
134      Y,  JUMP(5)
135          $ - 1, Z
136      Y,  S = CONSOLECLAIMED, Z
137      N,  JUMP(-8)
138          S = :M0[0]
139          CONSOLECLAIMED = S
140          S = USPECIALRM, P
141      N,  UCLAIMPUN = A
142          S = M[B - 5]
143          UCLAIMPLO = S
144          S = UCURRUNNUM
145          URUNNUM = S
146          S = 1
147          UCURRUNNUM + S
148          UKILLVAR = S
149          UTREXV
150          G = UWHO
151          UPROGID = G
152          CHEEN
153          UTELE2
154          UPOOL - S
155          SUBCD(:UCONEREEVWHITE)
156          DO(MD[- 1])
157          B = MA[0]
158          SE33
159          UTELE5
160  UNAC(23)                                " HEAD NOT ACCEPTABLE
161          G = UM36
162          SUBCD(:CPRINTSTRING)
163          G = UWHO
164          SUBCD(:CPRINTSTRING)
165          G = MT[14]
166          SUBCD(:CPRINTSTRING)            " NOT ACCEPTABLE
167          G = MT[13]
168          SUBCD(:CPRINTSTRING)            " RU
169          SUBCD(:CPRINTDEC)               " DESIRED NUMBER OF PUNCHES
170          G = MT[11]
171          SUBCD(:CPRINTSTRING)            " PLO
172          A = M[B - 5]
173          SUBCD(:CPRINTDEC)               " DESIRED NUMBER OF PLOTTERS
174          G = MT[8]
175          SUBCD(:CPRINTSTRING)            " MT
176          A = M[B - 4]
177          SUBCD(:CPRINTDEC)               " DESIRED NUMBER OF TAPES
178          SE33
```

UNAC not acceptable on account of claim of too many punchers, plotters, magnetic tapes etc. the operator may try this tape again at a later moment.

} segment transition, this performs a "jump" to invariant address UTELE5

```
179         UTELE3
180         UINVMINACC
181         UINVMRU
182         UINVMRLO
183         UINVMT
184 UINVMT(2)
185         '026 035 135'
186         '776 000 000'

187 UNINT(12)                       " HEAD NOT INTERPRETABLE
188         SUBCD(:UCLAIMCONRED)
189         G = GL18
190         G = UM1[UG]             " NONSENS ON RE A/B
191         SUBCD(:CPRINTSTRING)
192         G = MT[6]
193         SUBCD(:CPRINTSTRING)
194         SUBCD(:UCONFREEWHITE)   " ENTRY POINT NOT ACCEPTABLE
195         DO(MD[-1])
196         CHEEN
197         UTARESKIP
198         SUBCD(:UTEXTRM)
199         UINVMNONSENS
200 UTELE0(5)
201         S = - 0
202         GL27 = S
203         G = UBLOCKSTR
204         G = MG[7]
205         CTERUG
206 UTELE1(14)
207         CHEEN
208         USP
209     U,  S - 87, Z
210     N,  SUBCD(:CTERUGADRES)
211         CHEEN
212         USP
213     U,  S - 9, P
214     Y,  JUMP(4)
215         GL27 = S
216         CHEEN
217         USP
218         CTERUG
219         SE33
220         UNINT
221 UTELE2(10)
222         S =:M0[0]
223         SUBCD(:UPROGNAME)
224         G = MT[6]
225         SUBCD(:CPRINTSTRING)
226         S = USTANDARDNEED
227     U,  S - UPOOL, P
228     Y,  S = UPOOL
229         GL16 = S
230         CTERUG
231         UINVMBEGIN
232 UTELE3(7)
233         G = MT[5]
234         SUBCD(:CPRINTSTRING)
235         A = GL27
236         SUBCD(:CPRINTDEC)
237         SE33
```

```
238          UNINT(6)
239          UINVMTELE
240  UTELE5(18)
241          UHEAR
242          $ = :M0[0]
243          $ - CONSOLECLAIMED, Z
244      N,  JUMP(12)
245          $ = 2
246          UARCKB[1] = $
247          DO(UINTRC[4])
248      U,  $ - UARCKB[1], Z
249      N,  JUMP(- 4)
250          B + 2
251          DOS(PSE76)
252          0
253          119
254          $ = 13
255          SUBCD(:PSE73)
256          KARAKTERUIT
257          SE33
258          HALGOLBLOCK
259  UINVMTELE(2)
260          '014 030 027'
261          '135 776 000'

262  UINVBSEG(1)
263          UINVBSEG

264  UNEXTSYM(17)                    " DELIVER NEXT SYMBOL
265          B + 2
266          DOS(PSE76)
267          0
268          1
269          $ = 13
270          SE73
271          CSYMINSP
272          F - 1, Z
273      Y,  JUMP(- 9)
274          F + 1, Z
275      N,  JUMP(4)
276          CHEEN
277          SYM
278          G = S, P
279      Y,  SUBCD(:CTERUGADRES)     " SYMBOL IN S AND G
280          SE33
281          UNINT                   " NOT INTERPRETABLE

282  USR(5)                          " SKIP SPACES
283          CHEEN
284          UNEXTSYM
285      U,  $ - 93, Z
286      Y,  JUMP(- 4)
287          CTERUG

288  USRCR(5)                        " SKIP SPACES AND CARRIAGE RETURNS
289          CHEEN
290          USR
291      U,  $ - 119, Z
292      Y,  JUMP(- 4)
293          CTERUG
```

```
294 UTARESKIP(19)
295         S = UNUMTRRMSTR
296         MC[0] = S                   " STACK STREAM NUMBER
297         S = :MC[- 1]
298         B + 2
299         SUBC(:RSE78)                " FILL FORMAL INTEGER VARIABLE
300         S = 13
301         SE73
302         CSKIPRATA                   " SKIP UNTIL END OF TAPE
303         S = - 1
304         G = UBLOCKSTR
305         MG[9] = - S                 " INITIALIZE TAPE NUMBER
306         MG[21] = - E                " INITIALIZE UTRHANDLE[21]
307         S + MC[- 1], R
308     Y,  JUMP(- 13)                  " INITIALIZE NEXT STREAM
309         E = 1
310         UNITS = E                   " UNITS := 0; STEEK := 1
311         S = URROCTIME
312         GBILL = - S                 " INITIALIZE GBILL
313         CTERUG

314 UFILLSTRING(17)
315         A = M[B - 4]                " BUILDING UP WORD
316         LCA(9), R                   " WORD COMPLETED?
317         A '*' - 511
318         A + S
319         M[B - 4] = A                " NEW BUILDING UP WORD
320     Y,  M[B - 4] - A                " INITIALIZE BUILDING UP WORD
321     Y,  JUMP(4)
322         S - 510, Z                  " END OF STRING?
323     N,  SUBCD(:CTERUGADRES)
324         LCA(9), R                   " WORD COMPLETED?
325     N,  LCA(9)
326         G = A
327         A = 1
328         PLUSA(M[B - 2])             " INCREASE INVARIANT ADDRESS
329         MC[0] = A                   " STACK INVARIANT ADDRESS
330         SUBCD(:RSE50)               " ASSIGN
331         CTERUG

332 UINVMBEGIN(3)                       " BEGIN.
333         '013 016 020'
334         '022 027 130'
335         '776 000 000'

336 UINVMINACC(5)                       " INACCEPTABLE, B
337         '022 027 012'
338         '014 014 016'
339         '031 035 012'
340         '013 025 016'
341         '127 135 776'
342 UINVMNONSENS(4)                     " NONSENSE.
343         '027 030 027'
344         '034 016 027'
345         '034 016 130'
346         '776 000 000'
347 UINVMBU(2)                          " BUB
348         '031 036 135'
349         '776 000 000'
```

```
350 UINVMRLO(2)                          " RLOB
351         '031 025 030'
352         '135 776 000'
353 UJOB2(6)
354     U,  S - 131, Z
355     Y,  JUMP(2)
356         SE33
357         UJOB3
358         SE33
359         CORRECTION

360 HVERTALER1(1)
361         HVERTALER1

362 UINVEXIT(103)
363         UDEAE
364         S = UKILLCOUNTER, P
365     N,  UAMVAR = S

366     Y,  S = GL4                          " MAKE LINK INVARIANT
367     Y,  S '*' - 511
368     Y,  S - MS[0]
369     Y,  GL4 - S
370         G = - MA[- 1], P                 " S1S?
371     N,  JUMP(35)
372         S = MA[- 2]                      " INV. STARTING ADDRESS $1$
373         GL5 = S
374     U,  S = UKILLCOUNTER, P              " DYNAMIC ERROR?
375     N,  JUMP(4)
376         S '+' GL4
377         S '*' - 511, Z
378     N,  A = 3
379     N,  GOTO(:USTERR)                    " SYSTEM ERROR
380         S = MA[- 3]                      " PHYSICAL RETURN ADDRESS $1$
381         S '*' - 511
382         S - MS[0]
383         MIN$(MA[- 3])                    " INVARIANT RETURN ADDRESS $1$
384         S '+' GL6
385         S '*' - 511, Z
386     N,  A = 4
387     N,  GOTO(:USTERR)                    " SYSTEM ERROR
388         S = MA[- 3]
389         GL6 = S
390         S = GL5
391         RUS(9)
392         S '*' DRT[1]
393         G = MS[0], P                     " S1S PAGE IN CORE?
394     Y,  A = MG[3], P                     " S1S PAGE HOLY?
395     N,  A = 5
396     N,  GOTO(:USTERR)                    " SYSTEM ERROR
397         SUBCD(:URROEANATION)
398         DO(MD[-1])
399         S = 1                            " REMOVE $1$ LEVEL
400         MD[- 1] - S
401         S = M[63]
402         GL7 = S
403         A = - MA[- 1]
404         S = MA[- 3]
405         M[63] = S
406         JUMP(9)
```

final rites of a program in a PM

↓ tests for correct value of a number of status variables; if not correct, a system error is signalled in USTERR

Some other variables are reset

the stack is collapsed

post mortem information is dumped etc. etc.

```
407          $ = UKILLCOUNTER, P
408      Y,  $ = UTRANSTIME, P

409      N,  JUMP(5)
410          $ = GL4
411          $ '+' GL6
412          $ '*' - 511, Z
413      N,  A = 6
414      N,  GOTO(:USTERR)                    " SYSTEM ERROR
415          GL5 = $                          " ZERO INDICATION NO $I$
416          $ = GL6
417          RUS(9)
418          $ '*' DRT[1]
419          G = MS[0], P                     " PAGE IN CORE?
420      Y,  A = MG[3], P                     " PAGE HOLY?
421      N,  A = 7
422      N,  GOTO(:USTERR)                    " SYSTEM ERROR
423          SUBCD(:UBROKANATION)
424      U,  $ - :USVLIB[- 1], E
425      U,  $ - :USVLIB[UNUMSVLIB - 1], E    " NOT IN LIBRARY
426      N,  JUMP(4)
427          G = MS[0]
428          G = MG[3], P                     " STILL HOLY?
429      Y,  A = 8
430      Y,  GOTO(:USTERR)                    " SYSTEM ERROR
431          $ = GL5, Z                       " NO $I$?
432      Y,  JUMP(9)
433          RUS(9)
434          $ '*' DRT[1]
435      U,  $ - :USVLIB[- 1], E
436      U,  $ - :USVLIB[UNUMSVLIB - 1], E    " NOT IN LIBRARY?
437      N,  JUMP(4)
438          G = MS[0]
439          G = MG[3], P                     " STILL HOLY?
440      Y,  A = 9
441      Y,  GOTO(:USTERR)
442          DO(MD[- 1])
443          F = -0
444          GIP = F                          " RESET STANDARD
445          F = MT[5]
446          FLV = F
447          $ = :ERV
448          $ + DRT[13]
449          FCV = $
450          JUMP(2)
451          SUBCD(:PSE57)
452      U,  $ = 0
453          F = :UEXSEM
454          UV                               " V(EXSEM)
455          $ = UKILLCOUNTER, P              " DYNAMIC ERROR?
456      N,  F = GL24                         " SAVE NUMBER OF SEGMENTS
457      N,  GL3 = F                          " AND STACK PAGES
458      N,  JUMP(18)
459          B = MA[0]
460          SUBCD(:UMORTAL)
461          CHEEN
462          URPRINTERR                       " DYNAMIC ERROR IN PRINTER STREAM
463          UDEAF
464          $ = 5                            " NEW KILLVAR VALUE
```

```
465          JUMP(10)                          " MERGE WITH NORMAL EXIT
466 UNORMALEXIT(247)                           " NORMAL PROGRAM EXIT
467          UDEAF
468          A = UKILLVAR
469          A - 1, Z                          " KILLVAR = 1?
470      Y,  A = UKILLCOUNTER, Z                " KILLCOUNTER ZERO?
471      N,  A = 10
472      N,  GOTO(:USTERR)                     " SYSTEM ERROR
473          A = UWAMR[1], Z                   " WHITE?
474      N,  A = 11
475      N,  GOTO(:USTERR)                     " SYSTEM ERROR
476          S + 6                             " NEW KILLVAR VALUE
477          UKILLVAR = S                      " ENTRY POINT DYNAMIC ERROR
478          S = M0[- 3]                       " ENTRY POINT OPERATOR EXIT
479          M[63] = S                         " RESET D
480          S = M0[1]
481          MD[- 1] = S                       " FILL BLOCK HEIGHT ZERO
482          B = M0[0]                         " B ON WP
483          F = :M0[0]                        " G ON TARGET LRP
484          SUBCD(:SCL)
485          DO(MD[- 1])
486          CHEEN
487          ORSMUK2
488          GL18 = G
489          SUBCD(:UIMMORTAL)
490          S = UKILLVAR
491          MC = S
492          B +-2
493          S = 9
494          SE73
495          JTARESERE
496          S = MC[- 1]
497          UKILLVAR = S
498          SUBCD(:UMORTAL)

499          S = GL16
500          GL16 - S
501          UPOOL + S                         " NUMBER OF NEEDED STACK PAGES I= - 0
502          CHEEN
503          ULASTFORM                         " LAST PRINTER FORM
504          S = ULRSTREAMS                    " FINISH PRINTER STREAM
505          S = MS[0]
506          UBLOCKSTR = S
507      U,  S = MS[6], P                      " DOCUMENT UNDER CONSTRUCTION?
508      Y,  S = MS[15]                        " NUMBER OF PRINTER FORMS
509      N,  S = - 0
510          GL19 = S
511      N,  JUMP(3)
512          CHEEN
513          CENDLRDOC
514          DO(MD[- 1])
515          S = UNUMTRPMSTR                   " FINISH PUNCH STREAMS
516          MC[0] = S
517          S = :MC[- 1]
518          B + 2
519          SUBC(:RSE78)
520          S = 13
521          SE73
522          CENDTRDOC!
523          G = UBLOCKSTR
```

```
524          S = - 1
525          MG[9] = -S                        " RESET DOCUMENT NUMBER
526          S + MC[- 1], Z
527      N,  JUMP(- 12)
528          B + 2
529          S = 9
530          SE73
531          CENDRLDOC
532          G = URLSTREAMS
533          G = MG[0]
534          S = 1
535          MG[9] = S
536          UDEAF
537          UTREXPROOD
538          S = ULOANRUN, Z
539      Y,  S = ULOANRLO, Z
540      N,  A = 12
541      N,  GOTO(:USTERR)                     " SYSTEM ERROR
542          UCLAIMRUN = S
543          UCLAIMRLO = S
544          SUBCD(:UDISCRLU)                  " DISCONNECT PLUNCHER
545          UTREXVWIT
546          DO(MD[- 1])
547          CHEEN
548          UTARESKIP                         " SKIP UNTIL END OF TAPE
549          S = UKIND
550          S - 2, B
551      Y,  JUMP(54)
552          G = :URUNSEM                      " EXCLUSION SEMAPHORE FOR
553          UR                                " ACCES TO RUNSTATISTICS
554          UDEAF
555          SUBCD(:UCLAIMCONRED)
556          S = URUNAAN                       " INV. ADDRESS LAST FILLED PLACE
557      U,  S - MT[9], Z                      " IS THIS THE LAST PLACE IN THE RUN ARRAY?
558          G = MT[9]
559      Y,  SUBCD(:CPRINTSTRING)              " RUN STATISTICS DESTROYED
560      Y,  S = MT[8]                         " JUST BEFORE FIRST PLACE IN THE RUN ARRAY
561      Y,  URUNAAN = S
562      U,  S - MT[7], Z                      " WARNING POINT
563          G = MT[7]
564      Y,  SUBCD(:CPRINTSTRING)              " ASK FOR RUNSTATISTICS
565          SUBCD(:UCONFREEWHITE[1])
566          JUMP(5)
567          UINVRUN1[URUNLENGTH - 1]          " UINVRUNLAST
568          UINVDESTROYED
569          UINVRUN1[- 1]                     " UINVRUNNULLTH
570          UINVRUN1[URUNSIGNAL - 1]
571          UINVRUNSTAT

572          G = URUNNUM                       " SERIAL NUMBER
573          SUBCD(:MT[19])
574          A = - 0
575          MC[0] = A
576          A + UPROGID, Z                    " ADD INV. ADDRESS FIRST WORD NAME
577          SUBCD(:RSE39)                     " RIGHT SUBSCRIPTED INTEGER
578          SUBCD(:MT[14])
579          A = MC[- 1]
580          A + 1
581      U,  A - UNAMEWORDS, Z                 " TRANSPORT NAME FINISHED?
582      N,  JUMP(- 8)
```

```
583          G = UKILLVAR                    " DEATH CERTIFICATE
584          SUBCD(:MT[8])
585          G = GL19                        " NUMBER OF PRINTER FORMS
586          SUBCD(:MT[6])
587          G = UTRANSTIME                  " TRANSLATION TIME
588          SUBCD(:MT[4])
589          G = GL18                        " TOTAL TIME
590          G - UTRANSTIME                  " SUBTRACT TRANSLATION TIME
591          SUBCD(:MT[1])
592          JUMP(5)
593          S = 1
594          PLUS$(UBUNAAN)
595          MC[0] = S                       " INVARIANT ADDRESS TO BE FILLED
596          $UBCD(:PSE50)                   " ASSIGN
597          GOTOR(MC[- 1])
598          G = UBLSTREAMS
599          G = MG[0]
600          S = MG[22]                      " NUMBER OF PLOTTERSTEPS
601          MG[22] - S
602          G = S
603          SUBCD(:MT[- 11])
604          F = :UBUNSEM
605          UV
606          UDEAF
607          SUBCD(:UCLAIMCONRED)
608          CHEEN
609          UTELE4
610          G = MT[7]
611          SUBCD(:CBRINTSTRING)            " TIMEB
612          A = GL18
613          SUBCD(:CBRINTDEC)               " TOTAL TIME
614          G = MT[4]
615          SUBCD(:CBRINTSTRING)            " END.
616          SUBCD(:UCONFREEVHITE)

617          JUMP(2)
618          UINVBRTIME
619          UINVBREND
620          S = M0[- 28], Z
621     Y,   S = UKILLCOUNTER, Z
622     Y,   S = UAMTELREQ, Z
623     Y,   S = UAMVAR, Z
624     Y,   S = UAMSEM, Z
625     Y,   S = UAMSEM[1], Z
626     Y,   S = UAMSEM[2], Z
627     Y,   S = USCAMSEM, Z
628     Y,   S = USCAMSEM[1], Z
629     Y,   S = USCAMSEM[2], Z
630     Y,   S = UHAMR, Z
631     Y,   S = UHAMR[1], Z
632     Y,   S = M0[2], Z
633     Y,   S = GL24, Z
634     Y,   S = GL25, Z
635     Y,   S = UKILLSEM, Z
636     Y,   S = UKILLSEM[1], Z
637     Y,   S = UKILLSEM[2], Z
638     Y,   S = UDOCRUN, Z
639     Y,   S = UDOCRLO, Z
640     N,   A = 13
641     N,   GOTO(:USTERR)
```

more testing for status variables

```
642          A = - 0
643          MC[0] = A
644          A + UTRSTREAMS
645          A = MA[0]
646          S = MA[0], Z
647     Y,   S = MA[1], Z
648     Y,   S = MA[2], Z
649     Y,   S = MA[3], Z
650     Y,   S = MA[6], P
651     Y,   S = - MA[7], P
652     Y,   S = MA[8], Z
653     Y,   S = MA[9]
654     Y,   S - 1, Z
655     N,   A = 14
656     N,   GOTO(:USTERR)
657          A = MC[- 1]

658          A + 1
659     U,   A - UNUMTRMSTR, Z
660     N,   JUMP(- 18)
661          A = - 0
662          MC[0] = A
663          A + UTRSTREAMS
664          A = MA[0]
665          S = - MA[6], P
666     Y,   S = MA[10], Z
667     Y,   S = MA[11], Z
668     Y,   S = MA[12], Z
669     Y,   S = MA[17], Z
670     N,   A = 15
671     N,   GOTO(:USTERR)
672          A = MC[- 1]
673          A + 1
674     U,   A - UNUMTRRMSTR, Z
675     N,   JUMP(- 14)
676          A = - 0
677          MC[0] = A
678          A + URLSTREAMS
679          A = MA[0]
680          S = - MA[6], P
681     Y,   S = MA[10], Z
682     Y,   S = MA[11], Z
683     Y,   S = MA[12], Z
684     Y,   S = MA[17], Z
685     N,   A = 16
686     N,   GOTO(:USTERR)
687          A = MC[- 1]
688          A + 1
689     U,   A - UNUMRLRMSTR, Z
690     N,   JUMP(- 14)
691          S = FLV
692          S - MT[18], Z
693     Y,   S = FRV
694     Y,   S - MT[17], Z
695     N,   JUMP(5)
696          S = FCV
697     U,   S - MT[15], Z
698     N,   S - :FRV
699     N,   S - DRT[13], Z
700     Y,   P = GIP, Z
```

```
701      N,  A = 17
702      N,  GOTO(:USTERR)
703          F = - 0
704          GL23 = G
705          UTRANSTIME = G
706          UKILLVAR = G
707          URUNNUM = G
708          $ = CMNN
709          URPROGID = $
710          $UBCD(:UTEXTRM)
711          $UBCD(:RSE57)
712      U,  $ = 0
713          DO(FRV)
714 UINVSESEG1(1)
715          UINVSESEG1
716 UINVSESEG2(1)
717          UINVSESEG2
718 UTELE4(5)
719          $ = :M0[0]
720      U,  $ - CONSOLECLAIMED, Z
721      Y,  CONSOLECLAIMED -$
722          SUBCD(:URPROGNAME)
723          CTERUG
724 UF268(22)
725          '002 006 010'
726          '130 135 012'
727          '012 027 033'
728          '030 016 031'
729          '135 037 012'
730          '027 135 016'
731          '016 027 135'
732          '014 030 027'
733          '034 030 025'
734          '016 135 031'
735          '033 030 014'
736          '016 015 036'
737          '033 016 135'
738          '043 030 027'
739          '015 016 033'
740          '135 037 030'
741          '030 033 012'
742          '017 020 012'
743          '012 027 015'
744          '016 135 014'
745          '025 012 022'
746          '026 776 000'
747 UTIJD(13)
748          A = 1
749          $ = 4
750          $E18
751          $ = MA[-6]
752      U,  $ - 9,Z
753      N,  $UBCD(:RSE19)
754          CHEEN
755          UCALCULATION
756          $ = 21
757          GBILL + $,R
758          $E115
759          F * 100
760          $E71
```

```
761 URRINTERR(183)
762         S = MC[- 1]
763         GL18 = S                    " SAVE INV. RETURN ADDRESS
764         S = 25
765         $UBCD(:PSE1)                " ANONYMOUS RESERVATION
766         S = GL18
767         MC[0] = S                   " STACK INVARIANT RETURN ADDRESS
768         B + 2
769         DOS(PSE76)
770         0
771         3
772         S = 13
773         SE73
774         CARRIAGE
775         S = UKIND
776         S - 1, Z                    " ALGOL?
777     N,  JUMP(9)
778         S = UTRANSTIME, Z           " INDEX OUT OF BOUNDS IN
779     Y,  S = GL3                     " ONE DIMENSIONAL ARRAY
780     Y,  S - 205, Z                  " DURING TRANSLATION ?
781     Y,  S = 199
782     Y,  GL3 = S                     " PROGRAMMA TE LANG
783     Y,  JUMP(3)
784         S = GL4
785         CHEEN
786         UEOUTREGEL
787         A = GL3                     " FAILLURE NUMBER
788         A - 199
789         A + MT[1], Z
790         JUMP(1)
791         UEAILL                      " INV. ADDRESS FIRST WORD FAILLURE ARRAY
792         SUBCD(:PSE39)
793         B + 2
794         SUBC(:PSE112)
795         S = 13
796         SE73
797         CPRINTTEXT
798         S = UKIND
799         S - 1, Z                    " ALGOL?
800     N,  SUBCD(:CTERUGADRES)
801         S = UTRANSTIME, Z           " TRANSLATION ERROR?
802     N,  JUMP(6)
803         CHEEN
804         UCALCULATION
805         UTRANSTIME = G
806         S = 2
807         SE33
808         UNORMALEXIT
809         B + 2
810         DOS(PSE76)

811         0
812         3
813         S = 13
814         SE73
815         CARRIAGE
816         G = MT[1]
817         JUMP(1)
818         UINVKETTING                 " DYNAMISCHE B KETTING B
```

printing of dynamic chain of procedure activations on occurrence of a dyn. error.

C1063.3
Part 3

```
819         B + 2
820         SUBC(:RSE112)
821         $ = 13
822         SE73
823         CRRINTTEXT
824         GL4 = A                          " LRR LEVEL TO BE EXAMINED
825         $ = GL5, Z                       " NO $I$?
826     N,  $ = GL7                          " D FROM $I$
827     N,  $ - M[63], Z                     " IS THIS THE RETURN D, BLOCK?
828     N,  JUMR(32)
829         G = GL4
830         $ = MG[- 1]                      " RETURN LRR
831         G = MG[- 3]                      " CURRENT D
832         G - M$[- 3], Z                   " CURRENT D = RETURN D, BLOCK?
833     Y,  GL4 = $                          " NEXT LEVEL
834     Y,  JUMR(- 6)
835         G = GL4
836         $ = MG[- 5]                      " INV, STARTING ADDRESS
837     U,  $ - MT[1], Z
838         JUMR(1)
839         UINVEXECUTION                    " INV, STARTING ADDRESS PROCEDURE EXECUTION
840     N,  JUMR(11)
841         $ = GL3, R                       " DYNAMIC CHAIN EMPTY?
842     N,  SUBCD(:CTERUGADRES)
843         G = MT[1]
844         JUMR(1)
845         UINVLEEG                         " LEEG.
846         B + 2
847         SUBC(:RSE112)
848         $ = 13

849         SE73
850         CRRINTTEXT
851         CTERUG
852         GL5 = $                          " SAVE INV, STARTING ADDRESS
853         $ = MG[- 7]
854         GL6 = $                          " SAVE INV, RETURN ADDRESS
855         $ = MG[- 1]                      " RETURN LRR
856         GL4 = $                          " LRR NEXT LEVEL
857         $ = MG[- 3]                      " CURRENT D
858     U,  $ - MG[- 4], Z                   " CURRENT D = CONTEXT D, PARAMETER?
859     Y,  GL7 = $                          " POSITIV INDICATION PARAMETER
860     N,  GL7 = - $                        " NEGATIV INDICATION PROCEDURE OR SWITCH
861         B + 2
862         DOS(RSE76)
863         0
864         2
865         $ = 13
866         SE73
867         CARRIAGE
868         $ = GL7, R                       " PARAMETER?
869     N,  JUMR(3)
870         G = MT[1]
871         JUMR(52)
872         UINVRARAMETER                    " RARAMETER B NLCR TEKST BBBBBB
873         UDEAF
874         G = M[63]
875         $ = M0[- 3]
876         M[63] = $
877         $ = :MNL                         " ADDRESS ARRAYWORD NAMELIST
```

```
878           GL18 = S
879           M[63] = G
880           F = - 1
881           MC[0] = - G                    " SAVE INDEX
882           S = GL18, Z
883           SUBC(MS[0])
884           SUBCD(:RSE39)                  " PARAMETERCODE IN G
885           S = - G, Z
886      Y,   A = 17

887      Y,   GOTO(:USTERR)                  " PROCEDURE OR SWITCH NOT IN NAMELIST
888           S '*' 1024, Z                  " PROCEDURE OR SWITCH?
889      N,   JUMP(7)
890           GL19 = G                       " SAVE PARAMETERCODE
891           G = - M[B - 1]
892           F = 1
893           S = GL18, Z
894           SUBC(MS[0])
895           SUBCD(:RSE39)                  " INV, STARTING ADDRESS IN G
896           G - GL5, Z                     " IS IT THE PROCEDURE LOOKED FOR?
897           G = - MC[- 1]
898      N,   F - 3
899      N,   JUMP(- 19)                     " EXAMINE NEXT ENTRY IN NAMELIST
900           F + 1
901           S = GL18, Z
902           SUBC(MS[0])
903           SUBCD(:RSE39)                  " IDENTIFIER NUMBER IN G
904           GL3 = G                        " SAVE IDENTIFIER NUMBER
905           S = GL19                       " PARAMETERCODE
906           S '*' 16, Z                    " PROCEDURE?
907      Y,   F = MT[1]
908      N,   F = MT[1]
909           JUMP(2)
910           UINVPROCEDURE                  " PROCEDURE B
911           UINVSWITCH                     " SWITCH BBBB
912           B + 2
913           SUBC(:RSE112)
914           S = 13
915           SE73
916           CPRINTTEXT
917           B + 2
918           S = 9
919           SE73
920           UIDPRINT
921           G = MT[1]
922           JUMP(1)

923           UINVDECLARATIE                 " DECLARATIE B
924           B + 2
925           SUBC(:RSE112)
926           S = 13
927           SE73
928           CPRINTTEXT
929           S = GL5                        " INV, STARTING ADDRESS
930           CHEEN
931           UPRINTREGEL[1]
932           G = MT[1]
933           JUMP(1)
934           UINVCALL                       " CALL BBBBBBB
935           B + 2
```

```
936           $UBC(:RSE112)
937           $ = 13
938           SE73
939           CRRINTTEXT
940           S = GL6
941           CWEEN
942           URRINTREGEL
943           GL3 = - B                          " INDICATION CHAIN NOT EMPTY
944           JUMP(- 116)
945   UFOUTREGEL(12)
946           G = UTRANSTIME, Z
947       N,  JUMP(8)
948           S = GL3
949       U,  S - 243, R
950           S - 246, E
951       Y,  JUMP(4)
952           S = MREGELDEL
953           GL19 = S
954           SE33
955           URRINTREGEL[64]
956           SE33
957           URRINTREGEL

958   URRINTREGEL(89)
959           S - 1

960           GL18 = S                           " SAVE INVARIANT ADDRESS
961           G = UTRANSTIME, Z                  " FOUT IN VERTALER?
962           G = MT[1]
963           JUMP(1)
964           UINVVERTALER                       " REGEL BBBB VERTALER
965       Y,  JUMP(8)
966           RUS(9)
967           $ '*' DRT[1]
968       U,  S - :USVLIB[- 1], E
969       U,  S - :USVLIB[UNUMSVLIB - 1], E      " NOT IN LIBRARY?
970       Y,  JUMP(9)
971           G = MT[1]
972           JUMP(1)
973           UINVBIBLIOTHEEK                    " REGEL B BIBLIOTHEEK
974           B + 2
975           $UBC(:RSE112)                      " FILL FORMAL STRING
976           S = 13
977           SE73
978           CRRINTTEXT
979           JUMP(63)
980           UDEAF
981           S = M[63]
982           G = MO[- 3]
983           M[63] = G
984           G = WOBTEXT                        " ARRAYWORD OBJECTTEXT
985           M[63] = S
986           A = - 0
987           G = MG[- 2], Z                     " LAST LIST EXAMINED
988       Y,  A = 18
989       Y,  GOTO(:USTERR)                      " INV. ADDRESS NOT IN OBJECTTEXT
990           A + MG[- 1]                        " SUM CRITERIA
991           S = MG[- 3]
992           S - GL18, R                        " INV. ADDRESS < POSITION + CRITERIUM?
993       N,  JUMP(- 7)
```

```
 994      U,  S - MG[- 1], P                  " INV. ADDRESS < POSITION
 995      Y,  JUMP(- 9)
 996          S - A
 997          GL18 = - S                      " SAVE ORDINAL NUMBER
 998          F = - 1                         " GL19 := PREVIOUS LINE NUMBER := - 0
 999          F = 0
1000          GL20 = G

1001          G = M[63]
1002          S = MO[- 3]
1003          M[63] = S
1004          S = :HRL, Z
1005          M[63] = G
1006          G = - GL20
1007          F - 1
1008          GL20 = - G                      " INCREASE CURRENT LINE NUMBER
1009          SUBC(MS[0])
1010          SUBCD(:RSE39)                   " NEW VALUE IN G
1011          S = G
1012      U,  S + 2, Z
1013      Y,  JUMP(- 13)
1014      U,  S + 1, Z
1015      Y,  A = 19
1016      Y,  GOTO(:USTERR)
1017          G - GL18
1018          F - 0, P
1019      N,  S = GL20
1020      N,  GL19 = S
1021      N,  JUMP(- 21)
1022          DO(MD[- 1])
1023          B + 2
1024          G = MT[1]
1025          JUMP(1)
1026          UINVREGEL                       " REGEL B
1027          SUBC(:RSE112)                   " FILL FORMAL STRING
1028          S = 13
1029          SE73
1030          CPRINTTEXT
1031          B + 2
1032          DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1033          0
1034          10
1035          DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1036          0
1037          0
1038          S = :GL19
1039          SUBC(:RSE78)                    " FILL FORMAL INTEGER VARIABLE
1040          S = 21
1041          SE73
1042          CABSEIXT

1043          B + 2
1044          S = 9
1045          SE73
1046          CNLCR
1047          CTERUG

1048 UIDPRINT(69)                            " SEE WIDPRINT
1049          A = 1
1050          S = 5
```

printing program identifies 3

found through
the arrays HIL (identifier list)
or HCL (corresponding list)
built up by the translator.

```
1051          $E18
1052          $ = 8
1053          $E0
1054          $ = 16
1055          $E1
1056          UDEAF
1057          G = M[63]
1058          $ = M0[- 3]
1059          M[63] = $
1060          $ = :HCL
1061          GL18 = $
1062          $ = :HIL
1063          GL19 = $
1064          M[63] = G
1065          G = - GL3
1066          $ = GL18, Z
1067          SUBC(MS)
1068          $E39
1069          MA[2] = G
1070          F = - 1
1071          G - GL3
1072          $ = GL18, Z
1073          SUBC(MS)
1074          $E39
1075          GL3 = G
1076          G = - MA[2]
1077          $ = GL19, Z
1078          SUBC(MS)
1079          $E39
1080          MA[3] = - G, P
1081          $ = G
1082          RUS(18)

1083          SUBC(:MT[19])
1084          RUS(12)
1085          $ '*' 63
1086          SUBC(:MT[14])
1087     N,   RUS(6)
1088     N,   $ '*' 63
1089     N,   SUBC(:MT[11])
1090     N,   $ '*' 63
1091     N,   SUBC(:MT[9])
1092          $ = 1
1093          PLUS$(MA[2])
1094          $ - GL3, Z
1095     N,   JUMP(- 20)
1096          $E6
1097          $ = 1
1098          GBILL + $, P
1099          $E115
1100          $E71
1101     U,   $ - 63, Z
1102     Y,   GOTO(MC[- 1])
1103          MA[4] = $
1104          B + 2
1105          $ = :MA[4]
1106          $E78
1107          $ = 13
1108          $E73
1109          CPRSYM
```

```
1110          S = MC[- 1]
1111          S '*' - DRT[16]
1112          MC =  S
1113          S = T
1114          S '*' DRT[16]
1115          M[B - 1] + S
1116          S = - MA[3]
1117          GOTOR(MC[- 1])
1118 UCALCULATION(14)
1119          G = UNITS
1120          F * 120
1121          MC = F
1122          G = URROCTIME
1123          G + GBILL
1124          F / 16 667
1125          F + MC[- 2]
1126          F + DRT[11]
1127          F - DRT[11], Z
1128      Y,  F = 1
1129      U,  S = M[57], Z
1130      Y,  JUMP(1)
1131          G = UCON3
1132          CTERUG

1133 UINVRUNSTAT(6)                          " RUNSTATISTICS
1134          '033 036 027'
1135          '034 035 012'
1136          '035 022 034'
1137          '035 022 014'
1138          '034 130 167'
1139          '776 000 000'
1140 UINVDESTROYED(9)                        " RUNSTATISTICS B DESTROYED
1141          '033 036 027'
1142          '034 035 012'
1143          '035 022 034'
1144          '035 022 014'
1145          '034 135 015'
1146          '016 034 035'
1147          '033 030 042'
1148          '016 015 130'
1149          '167 776 000'
1150 UINVRTIME(2)                            " TIME B
1151          '035 022 026'
1152          '016 135 776'
1153 UINVREND(2)                             " END.
1154          '016 027 015'
1155          '130 776 000'
1156 UINVTOTALETIJD(5)                       " TOTALE B TIJD BB
1157          '035 030 035'
1158          '012 025 016'
1159          '135 035 022'
1160          '023 015 135'
1161          '135 776 000'

1162 UINVVERTAALTIJD(5)                      " VERTAALTIJD BB
1163          '037 016 033'
1164          '035 012 012'
1165          '025 035 022'
1166          '023 015 135'
1167          '135 776 000'
```

```
1168 UINVEXECUTIETIJD(5)                    " EXECUTIETIJD B
1169         '016 041 016'
1170         '014 036 035'
1171         '022 016 035'
1172         '022 023 015'
1173         '135 776 000'
1174 UINVAANTALOVER(7)                      " AANTAL B OVER B STROOM
1175         '012 012 027'
1176         '035 012 025'
1177         '135 030 037'
1178         '016 033 135'
1179         '034 035 033'
1180         '030 030 026'
1181         '776 000 000'
1182 UINVGETALLEN(6)                        " GELEZEN B GETALLEN B
1183         '020 016 025'
1184         '016 043 016'
1185         '027 135 020'
1186         '016 035 012'
1187         '025 025 016'
1188         '027 135 776'

1189 UINVLAATSTOVER(7)                      " LAATST B OVER B STROOM
1190         '025 012 012'
1191         '035 034 035'
1192         '135 030 037'
1193         '016 033 135'
1194         '034 035 033'
1195         '030 030 026'
1196         '776 000 000'
1197 UINVGETAL(5)                           " GELEZEN B GETAL B
1198         '020 016 025'
1199         '016 043 016'
1200         '027 135 020'
1201         '016 035 012'
1202         '025 135 776'
1203 UINVPROGRAMMADOOR(13)                  " PROGRAMMA B DOOR B
1204         '031 033 030'                  " OPERATEUR B BEEINDIGD. NLCB
1205         '020 033 012'
1206         '026 026 012'
1207         '135 015 030'
1208         '030 033 135'
1209         '030 031 016'
1210         '033 012 035'
1211         '016 036 033'
1212         '135 013 016'
1213         '016 022 027'
1214         '015 022 020'
1215         '015 130 167'
1216         '776 000 000'

1217 UINVSTAREL(8)                          " AANTAL B STAPELPAGINA'S B
1218         '012 012 027'
1219         '035 012 025'
1220         '135 034 035'
1221         '012 031 016'
1222         '025 031 012'
1223         '020 022 027'
1224         '012 170 034'
1225         '135 776 000'
```

```
1226 UINVSEGMENTEN(6)                    " AANTAL B SEGMENTEN B
1227         '012 012 027'
1228         '035 012 025'
1229         '135 034 016'
1230         '020 026 016'
1231         '027 035 016'
1232         '027 135 776'
1233 UINVKETTING(7)                      " DYNAMISCHE B KETTING B
1234         '015 042 027'
1235         '012 026 022'
1236         '034 014 021'
1237         '016 135 024'
1238         '016 035 035'
1239         '022 027 020'
1240         '135 776 000'
1241 UINVLEEG(2)                         " LEEG.
1242         '025 016 016'
1243         '020 130 776'
1244 UINVPARAMETER(8)                    " PARAMETER B NLGB TEKST BBBBBB
1245         '031 012 033'
1246         '012 026 016'
1247         '035 016 033'
1248         '135 167 035'
1249         '016 024 034'
1250         '035 135 135'
1251         '135 135 135'
1252         '135 776 000'

1253 UINVPROCEDURE(4)                    " PROCEDURE BB
1254         '031 033 030'
1255         '014 016 015'
1256         '036 033 016'
1257         '135 135 776'
1258 UINVSWITCH(4)                       " SWITCH BBBBB
1259         '034 040 022'
1260         '035 014 021'
1261         '135 135 135'
1262         '135 135 776'
1263 UINVDECLARATIE(5)                   " NLGB DECLARATIE B
1264         '167 015 016'
1265         '014 025 012'
1266         '033 012 035'
1267         '022 016 135'
1268         '776 000 000'
1269 UINVCALL(4)                         " CALL BBBBBBB
1270         '014 012 025'
1271         '025 135 135'
1272         '135 135 135'
1273         '135 135 776'
1274 UINVVERTALER(6)                     " REGEL BBBB VERTALER
1275         '033 016 020'
1276         '016 025 135'
1277         '135 135 135'
1278         '037 016 033'
1279         '035 012 025'
1280         '016 033 776'
1281 UINVBIBLIOTHEEK(6)                  " REGEL B BIBLIOTHEEK
1282         '033 016 020'
1283         '016 025 135'
1284         '013 022 013'
```

```
1285          '025 022 030'
1286          '035 021 016'
1287          '016 024 776'

1288 UINVREGEL(3)                          " REGEL B
1289          '033 016 020'
1290          '016 025 135'
1291          '776 000 000'
1292 UINVRUN1(1)
1293          = 0
1294 UINVRUN2(1)
1295          = 0
1296 UINVRUN3(1)
1297          = 0
1298 UINVRUN4(1)
1299          = 0
1300 UINVRUN5(1)
1301          = 0
1302 UINVRUN6(1)
1303          = 0
1304 UINVRUN7(1)
1305          = 0
1306 UINVRUN8(1)
1307          = 0

1308 ULASTFORM(249)
1309          S = UKIND
1310          S = 1, Z                     " ALGOL?
1311          G = ULRSTREAMS
1312          G = MG[0]
1313     N,   S = MG[6], B                 " DOCUMENT UNDER CONSTRUCTION?
1314     N,   S = UKILLVAR
1315     N,   S = 4, Z                     " THROWN OUT BY OPERATOR?
1316     N,   SUBCD(:CTERUGADRES)
1317          B + 2
1318          S = 9
1319          SE73
1320          CNEWPAGE
1321          B + 2
1322          S = 9
1323          SE73
1324          CNLCR
1325          S = UKIND
1326          S - 1, Z
1327     Y,   JUMP(21)                     " ALGOL?
1328          B + 2
1329          G = MT[1]
1330          JUMP(1)
1331          UINVTOTALETIJD               " TOTALE B TIJD BB
1332          SUBC(:RSE112)                " FILL FORMAL STRING
1333          S = 13
1334          SE73
1335          CPRINTTEXT
1336          B + 2
1337          DOS(RSE76)                   " FILL FORMAL INTEGER CONSTANT
1338          0
1339          6
1340          DOS(RSE76)                   " FILL FORMAL INTEGER CONSTANT
1341          0
1342          0
```

The effect of lastform is found on 685 of this program print out

```
1343        $ = :GL18                       " TOTALE TIJD
1344        $UBC(:RSE78)                    " FILL FORMAL INTEGER VARIABLE
1345        $ = 21

1346        SE73
1347        CABSEIXT
1348        JUMP(146)
1349        B + 2
1350        G = MT[1]
1351        JUMP(1)
1352        UINVVERTAALTIJD                 " VERTAALTIJD BB
1353        $UBC(:RSE112)                   " FILL FORMAL STRING
1354        $ = 13
1355        SE73
1356        CPRINTTEXT
1357        B + 2
1358        DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1359        0
1360        6
1361        DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1362        0
1363        0
1364        $ = :UTRANSTIME                 " VERTAALTIJD
1365        SUBC(:RSE78)                    " FILL FORMAL INTEGER VARIABLE
1366        $ = 21
1367        SE73
1368        CABSEIXT
1369        B + 2
1370        $ = 9
1371        SE73
1372        CNLCR
1373        B + 2
1374        G = MT[1]
1375        JUMP(1)
1376        UINVEXECUTIETIJD                " EXECUTIETIJD B
1377        SUBC(:RSE112)                   " FILL FORMAL STRING
1378        $ = 13
1379        SE73
1380        CPRINTTEXT
1381        B + 2
1382        DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1383        0
1384        6

1385        DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1386        0
1387        0
1388        G = GL18
1389        G - UTRANSTIME
1390        GL19 = G
1391        $ = :GL19                       " EXECUTIETIJD
1392        SUBC(:RSE78)                    " FILL FORMAL INTEGER VARIABLE
1393        $ = 21
1394        SE73
1395        CABSEIXT
1396        $ = 1
1397        GL19 = $
1398        B + 2
1399        DOS(RSE76)                      " FILL FORMAL INTEGER CONSTANT
1400        0
```

```
1401          2
1402          $ = 13
1403          SE73
1404          CARRIAGE
1405          B + 2
1406          G = MT[1]
1407          JUMP(1)
1408          UINVAANTALOVER                  " AANTAL B OVER B STROOM
1409          SUBC(:PSE112)                   " FILL FORMAL STRING
1410          $ = 13
1411          SE73
1412          CPRINTTEXT
1413          B + 2
1414          DOS(PSE76)                      " FILL FORMAL INTEGER CONSTANT
1415          0
1416          1
1417          DOS(PSE76)                      " FILL FORMAL INTEGER CONSTANT
1418          0
1419          0
1420          S = :GL19
1421          SUBC(:PSE78)                    " FILL FORMAL INTEGER VARIABLE
1422          S = 21

1423          SE73
1424          CABSEIXT
1425          B + 2
1426          G = MT[1]
1427          JUMP(1)
1428          UINVGETALLEN                    " GELEZEN B GETALLEN B
1429          SUBC(:PSE112)                   " FILL FORMAL STRING
1430          S = 13
1431          SE73
1432          CPRINTTEXT
1433          B + 2
1434          DOS(PSE76)
1435          0
1436          10
1437          DOS(PSE76)
1438          0
1439          0
1440          S = UTRSTREAMS
1441          S + GL19
1442          S = MS[- 1]                     " :UTRHANDLE
1443          GL20 = S
1444          S + 21
1445          SUBC(:PSE78)                    " FILL FORMAL INTEGER VARIABLE
1446          S = 21
1447          SE73
1448          CABSEIXT
1449          S = GL20
1450          S = MS[21], Z                   " NUMBER OF READ NUMBERS ZERO?
1451      Y,  JUMP(39)
1452          B + 2
1453          S = 9
1454          SE73
1455          CNLCR
1456          B + 2
1457          G = MT[1]
1458          JUMP(1)
1459          UINVLAATSTOVER                  " LAATST B OVER B STROOM
```

```
1460          SUBC(:RSE112)

1461          S = 13
1462          SE73
1463          CRRINTTEXT
1464          B + 2
1465          DOS(RSE76)                    " FILL FORMAL INTEGER CONSTANT
1466          0
1467          1
1468          DOS(RSE76)                    " FILL FORMAL INTEGER CONSTANT
1469          0
1470          0
1471          S = :GL19
1472          SUBC(:RSE78)                  " FILL FORMAL INTEGER VARIABLE
1473          S = 21
1474          SE73
1475          CABSEIXT
1476          B + 2
1477          G = MT[1]
1478          JUMR(1)
1479          UINVGETAL                     " GELEZEN B GETAL B
1480          SUBC(:RSE112)                 " FILL FORMAL STRING
1481          S = 13
1482          SE73
1483          CRRINTTEXT
1484          B + 2
1485          S = GL20                      " UTRHANDLE
1486          S + 22
1487          SUBCD(:RSE77)                 " FILL FORMAL REAL VARIABLE
1488          S = 13
1489          SE73
1490          CRRNTX
1491          S = 1
1492          RLUSS(GL19)
1493          S - UNUMTRRMSTR, R
1494     N,   JUMR(- 97)
1495          S = UKILLVAR
1496          S - 4, Z
1497     N,   SUBCD(:CTERUGADRES)
1498          B + 2

1499          DOS(RSE76)                    " FILL FORMAL INTEGER CONSTANT
1500          0
1501          2
1502          S = 13
1503          SE73
1504          CARRIAGE
1505          B + 2
1506          G = MT[1]
1507          JUMR(1)
1508          UINVRROGRAMMADOOR             " RROGRAMMA B DOOR B ORERATEUR B BEEINDIGD. NLCR
1509          SUBC(:RSE112)                 " FILL FORMAL STRING
1510          S = 13
1511          SE73
1512          CRRINTTEXT
1513          B + 2
1514          G = MT[1]
1515          JUMR(1)
1516          UINVSTAREL                    " AANTAL B STARELRAGINA'S B
1517          SUBC(:RSE112)                 " FILL FORMAL STRING
```

```
1518          $ = 13
1519          $E73
1520          CPRINTTEXT
1521          B + 2
1522          DO$(P$E76)                    " FILL FORMAL INTEGER CONSTANT
1523          0
1524          3
1525          DO$(P$E76)                    " FILL FORMAL INTEGER CONSTANT
1526          0
1527          0
1528          $ = :GL4
1529          $UBC(:P$E78)                  " FILL FORMAL INTEGER VARIABLE
1530          $ = 21
1531          $E73
1532          CABSEIXT
1533          B + 2
1534          $ = 9
1535          $E73
1536          CNLCR

1537          B + 2
1538          G = MT[1]
1539          JUMP(1)
1540          UINVSEGMENTEN                 " AANTAL B $EGMENTEN B
1541          $UBC(:P$E112)
1542          $ = 13
1543          $E73
1544          CPRINTTEXT
1545          B + 2
1546          DO$(P$E76)
1547          0
1548          8
1549          DO$(P$E76)
1550          0
1551          0
1552          $ = :GL3
1553          $UBC(:P$E78)
1554          $ = 21
1555          $E73
1556          CABSEIXT
1557          CTERUG

1558 WE99(8)
1559 '011 011 130'
1560 '135 031 033'
1561 '030 020 033'
1562 '012 026 026'
1563 '012 135 035'
1564 '016 135 025'
1565 '012 027 020'
1566 '776 000 000'

1567 WE200(27)
1568 '002000000'                    " 200
1569 '130135022'
1570 '027135012'
1571 '033033012'
1572 '042135015'
1573 '016014025'
1574 '012033012'
```

↓ dyn. error message-strings

```
1575 '035022016'
1576 '135025030'
1577 '040016033'
1578 '013030036'
1579 '027015135'
1580 '020033030'
1581 '035016033'
1582 '135015012'
1583 '027135036'
1584 '031031016'
1585 '033013030'
1586 '036027015'
1587 '135142014'
1588 '017130033'
1589 '016037130'
1590 '033016031'
1591 '130135005'
1592 '130002130'
1593 '004130003'
1594 '130143776'

1595 WF201(11)
1596 '002000001'                " 201
1597 '130135020'
1598 '016015016'
1599 '014025012'
1600 '033016016'
1601 '033015135'
1602 '012033033'
1603 '012042135'
1604 '035016135'
1605 '020033030'
1606 '030035776'

1607 WF202(14)
1608 '002000002'                " 202
1609 '130135012'
1610 '033033012'
1611 '042135013'
1612 '042135037'
1613 '012025036'
1614 '016135027'
1615 '022016035'
1616 '135020016'
1617 '022026031'
1618 '025016026'
1619 '016027035'
1620 '016016033'
1621 '015776000'

1622 WF203(24)
1623 '002000003'                " 203
1624 '130135022'
1625 '027135012'
1626 '033033012'
1627 '042135015'
1628 '016014025'
1629 '012033012'
1630 '035022016'
1631 '135036031'
```

```
1632 '031016033'
1633 '013030036'
1634 '027015135'
1635 '031025036'
1636 '034135016'
1637 '016027135'
1638 '013036022'
1639 '035016027'
1640 '135022027'
1641 '035016020'
1642 '016033014'
1643 '012031012'
1644 '014022035'
1645 '016022035'
1646 '776000000'

1647 UE204(31)
1648 '002000004'                " 204
1649 '130135022'
1650 '027135034'
1651 '036013034'
1652 '014033022'
1653 '031035016'
1654 '015135037'
1655 '012033022'
1656 '012013025'
1657 '016135022'
1658 '034135021'
1659 '016035135'
1660 '012012027'
1661 '035012025'
1662 '135034036'
1663 '013034014'
1664 '033022031'
1665 '035034135'
1666 '030027020'
1667 '016025022'
1668 '023024135'
1669 '012012027'
1670 '135015016'
1671 '135015022'
1672 '026016027'
1673 '034022016'
1674 '135037012'
1675 '027135021'
1676 '016035135'
1677 '012033033'
1678 '012042776'

1679 UE205(20)
1680 '002000005'                " 205
1681 '130135034'
1682 '036013034'
1683 '014033022'
1684 '031035135'
1685 '013036022'
1686 '035016027'
1687 '135015016'
1688 '135020033'
1689 '016027043'
```

```
1690 '016027135'
1691 '142016016'
1692 '027135015'
1693 '022026016'
1694 '027034022'
1695 '030027012'
1696 '012025135'
1697 '012033033'
1698 '012042143'
1699 '776000000'

1700 VE206(20)
1701 '002000006'                " 206
1702 '130135034'
1703 '036013034'
1704 '014033022'
1705 '031035135'
1706 '013036022'
1707 '035016027'
1708 '135015016'
1709 '135020033'
1710 '016027043'
1711 '016027135'
1712 '142026016'
1713 '016033135'
1714 '015022026'
1715 '016027034'
1716 '022030027'
1717 '012012025'
1718 '135012033'
1719 '033012042'
1720 '143776000'

1721 VE207(22)
1722 '002000007'                " 207
1723 '130135034'
1724 '031016014'
1725 '022017022'
1726 '014012035'
1727 '022016135'
1728 '012033022'
1729 '035021026'
1730 '016035022'
1731 '014133135'
1732 '012014035'
1733 '036016025'
1734 '016135031'
1735 '012033012'
1736 '026016035'
1737 '016033135'
1738 '027022016'
1739 '035135035'
1740 '030016020'
1741 '016034035'
1742 '012012027'
1743 '776000000'

1744 VE208(21)
1745 '002000010'                " 208
1746 '130135034'
```

```
1747 '031016014'
1748 '022017022'
1749 '014012035'
1750 '022016135'
1751 '013030030'
1752 '025016012'
1753 '027133135'
1754 '012014035'
1755 '036016025'
1756 '016135031'
1757 '012033012'
1758 '026016035'
1759 '016033135'
1760 '027022016'
1761 '035135035'
1762 '030016020'
1763 '016034035'
1764 '012012027'
1765 '776000000'

1766 UE209(21)
1767 '002000011'                " 209
1768 '130135034'
1769 '031016014'
1770 '022017022'
1771 '014012035'
1772 '022016135'
1773 '014030026'
1774 '031025016'
1775 '041133135'
1776 '012014035'
1777 '036016025'
1778 '016135031'
1779 '012033012'
1780 '026016035'
1781 '016033135'
1782 '027022016'
1783 '035135035'
1784 '030016020'
1785 '016034035'
1786 '012012027'
1787 '776000000'

1788 UE210(24)
1789 '002001000'                " 210
1790 '130135034'
1791 '031016014'
1792 '022017022'
1793 '014012035'
1794 '022016135'
1795 '012033022'
1796 '035021026'
1797 '016035022'
1798 '014135012'
1799 '033033012'
1800 '042133135'
1801 '012014035'
1802 '036016025'
1803 '016135031'
1804 '012033012'
```

```
1805 '02601605'
1806 '016033135'
1807 '027022016'
1808 '035135035'
1809 '030016020'
1810 '016034035'
1811 '012012027'
1812 '776000000'

1813 UF211(23)
1814 '002001001'                " 211
1815 '130135034'
1816 '031016014'
1817 '022017022'
1818 '014012035'
1819 '022016135'
1820 '013030030'
1821 '025016012'
1822 '027135012'
1823 '033033012'
1824 '042133135'
1825 '012014035'
1826 '036016025'
1827 '016135031'
1828 '012033012'
1829 '026016035'
1830 '016033135'
1831 '027022016'
1832 '035135035'
1833 '030016020'
1834 '016034035'
1835 '012012027'
1836 '776000000'

1837 UF212(23)
1838 '002001002'                " 212
1839 '130135034'
1840 '031016014'
1841 '022017022'
1842 '014012035'
1843 '022016135'
1844 '014030026'
1845 '031025016'
1846 '041135012'
1847 '033033012'
1848 '042133135'
1849 '012014035'
1850 '036016025'
1851 '016135031'
1852 '012033012'
1853 '026016035'
1854 '016033135'
1855 '027022016'
1856 '035135035'
1857 '030016020'
1858 '016034035'
1859 '012012027'
1860 '776000000'

1861 UF213(21)
```

```
1862 '002001003'                " 213
1863 '130135034'
1864 '031016014'
1865 '022017022'
1866 '014012035'
1867 '022016135'
1868 '031033030'
1869 '014016015'
1870 '036033016'
1871 '133135012'
1872 '014035036'
1873 '016025016'
1874 '135031012'
1875 '033012026'
1876 '016035016'
1877 '033135027'
1878 '022016035'
1879 '135035030'
1880 '016020016'
1881 '034035012'
1882 '012027776'

1883 WE214(25)
1884 '002001004'                " 214
1885 '130135034'
1886 '031016014'
1887 '022017022'
1888 '014012035'
1889 '022016135'
1890 '012033022'
1891 '035021026'
1892 '016035022'
1893 '014135031'
1894 '033030014'
1895 '016015036'
1896 '033016133'
1897 '135012014'
1898 '035036016'
1899 '025016135'
1900 '031012033'
1901 '012026016'
1902 '035016033'
1903 '135027022'
1904 '016035135'
1905 '035030016'
1906 '020016034'
1907 '035012012'
1908 '027776000'

1909 WE215(24)
1910 '002001005'                " 215
1911 '130135034'
1912 '031016014'
1913 '022017022'
1914 '014012035'
1915 '022016135'
1916 '013030030'
1917 '025016012'
1918 '027135031'
1919 '033030014'
```

```
1920 '016015036'
1921 '033016133'
1922 '135012014'
1923 '035036016'
1924 '025016135'
1925 '031012033'
1926 '012026016'
1927 '035016033'
1928 '135027022'
1929 '016035135'
1930 '035030016'
1931 '020016034'
1932 '035012012'
1933 '027776000'

1934 VF216(24)
1935 '002001006'                " 216
1936 '130135034'
1937 '031016014'
1938 '022017022'
1939 '014012035'
1940 '022016135'
1941 '014030026'
1942 '031025016'
1943 '041135031'
1944 '033030014'
1945 '016015036'
1946 '033016133'
1947 '135012014'
1948 '035036016'
1949 '025016135'
1950 '031012033'
1951 '012026016'
1952 '035016033'
1953 '135027022'
1954 '016035135'
1955 '035030016'
1956 '020016034'
1957 '035012012'
1958 '027776000'

1959 VF217(20)
1960 '002001007'                " 217
1961 '130135034'
1962 '031016014'
1963 '022017022'
1964 '014012035'
1965 '022016135'
1966 '025012013'
1967 '016025133'
1968 '135012014'
1969 '035036016'
1970 '025016135'
1971 '031012033'
1972 '012026016'
1973 '035016033'
1974 '135027022'
1975 '016035135'
1976 '035030016'
1977 '020016034'
```

```
1978 '035012012'
1979 '027776000'

1980 VF218(20)
1981 '002001010'                " 218
1982 '130135034'
1983 '031016014'
1984 '022017022'
1985 '014012035'
1986 '022016135'
1987 '034040022'
1988 '035014021'
1989 '133135012'
1990 '014035036'
1991 '016025016'
1992 '135031012'
1993 '033012026'
1994 '016035016'
1995 '033135027'
1996 '022016035'
1997 '135035030'
1998 '016020016'
1999 '034035012'
2000 '012027776'

2001 VF219(20)
2002 '002001011'                " 219
2003 '130135034'
2004 '031016014'
2005 '022017022'
2006 '014012035'
2007 '022016135'
2008 '034035033'
2009 '022027020'
2010 '133135012'
2011 '014035036'
2012 '016025016'
2013 '135031012'
2014 '033012026'
2015 '016035016'
2016 '033135027'
2017 '022016035'
2018 '135035030'
2019 '016020016'
2020 '034035012'
2021 '012027776'

2022 VF220(36)
2023 '002002000'                " 220
2024 '130135022'
2025 '027135015'
2026 '016135025'
2027 '016017035'
2028 '135031012'
2029 '033035135'
2030 '025022034'
2031 '035135037'
2032 '012027135'
2033 '016016027'
2034 '135026036'
```

```
2035 '025035022'
2036 '031025016'
2037 '135012034'
2038 '034022020'
2039 '027026016'
2040 '027035135'
2041 '034035012'
2042 '035016026'
2043 '016027035'
2044 '135043022'
2045 '023027135'
2046 '015016135'
2047 '037012033'
2048 '022012013'
2049 '016025016'
2050 '027135027'
2051 '022016035'
2052 '135037012'
2053 '027135021'
2054 '016035043'
2055 '016025017'
2056 '015016135'
2057 '035042031'
2058 '016776000'

2059 VF221(30)
2060 '002002001'                " 221
2061 '130135022'
2062 '027135016'
2063 '016027135'
2064 '017030033'
2065 '135034035'
2066 '012035016'
2067 '026016027'
2068 '035135026'
2069 '030016035'
2070 '135015016'
2071 '135014030'
2072 '027035033'
2073 '030025025'
2074 '016015135'
2075 '037012033'
2076 '022012013'
2077 '025016135'
2078 '016016027'
2079 '135033016'
2080 '012025135'
2081 '030017135'
2082 '022027035'
2083 '016020016'
2084 '033135037'
2085 '012033022'
2086 '012013025'
2087 '016135043'
2088 '022023027'
2089 '776000000'

2090 VF222(42)
2091 '002002002'                " 222
2092 '130135021'
```

```
2093 '016035135'
2094 '033016034'
2095 '036025035'
2096 '012012035'
2097 '135037012'
2098 '027135015'
2099 '016135012'
2100 '017033030'
2101 '027015022'
2102 '027020135'
2103 '025022020'
2104 '035135013'
2105 '036022035'
2106 '016027135'
2107 '022027035'
2108 '016020016'
2109 '033135014'
2110 '012031012'
2111 '014022035'
2112 '016022035'
2113 '135142014'
2114 '017130033'
2115 '016037130'
2116 '033016031'
2117 '130135003'
2118 '130001130'
2119 '004130002'
2120 '127135003'
2121 '130005130'
2122 '004127135'
2123 '004130002'
2124 '130004127'
2125 '135004130'
2126 '007130003'
2127 '130001135'
2128 '016027135'
2129 '005130002'
2130 '130004130'
2131 '001130143'
2132 '776000000'

2133 VF223(18)
2134 '002002003'   " 223
2135 '130135031'
2136 '033030014'
2137 '016015036'
2138 '033016135'
2139 '012012027'
2140 '033030016'
2141 '031135026'
2142 '016035135'
2143 '037016033'
2144 '024016016'
2145 '033015135'
2146 '012012027'
2147 '035012025'
2148 '135031012'
2149 '033012026'
2150 '016035016'
2151 '033034776'
```

```
2152 WE224(19)
2153 '002002004'                " 224
2154 '130135040'
2155 '012012033'
2156 '015016135'
2157 '035030016'
2158 '024016027'
2159 '027022027'
2160 '020135012'
2161 '012027135'
2162 '012014035'
2163 '036016025'
2164 '016135031'
2165 '012033012'
2166 '026016035'
2167 '016033135'
2168 '030027026'
2169 '030020016'
2170 '025022023'
2171 '024776000'

2172 WE225(36)
2173 '002002005'                " 225
2174 '130135040'
2175 '012012033'
2176 '015016135'
2177 '035030016'
2178 '024016027'
2179 '027022027'
2180 '020135012'
2181 '012027135'
2182 '016016027'
2183 '135012033'
2184 '022035021'
2185 '026016035'
2186 '022034014'
2187 '021135012'
2188 '033033012'
2189 '042135016'
2190 '025016026'
2191 '016027035'
2192 '135040016'
2193 '020016027'
2194 '034135034'
2195 '031016014'
2196 '022017022'
2197 '014012035'
2198 '022016135'
2199 '014030026'
2200 '031025016'
2201 '041135012'
2202 '033033012'
2203 '042135027'
2204 '022016035'
2205 '135035030'
2206 '016020016'
2207 '034035012'
2208 '012027776'
```

```
2209 VE226(31)
2210 '002002006'                " 226
2211 '130135014'
2212 '030026031'
2213 '025016041'
2214 '016135012'
2215 '014035036'
2216 '016025016'
2217 '135031012'
2218 '033012026'
2219 '016035016'
2220 '033135022'
2221 '027135016'
2222 '041031033'
2223 '016034034'
2224 '022016135'
2225 '040016020'
2226 '016027034'
2227 '135034031'
2228 '016014022'
2229 '017022014'
2230 '012035022'
2231 '016135012'
2232 '033022035'
2233 '021026016'
2234 '035022014'
2235 '135027022'
2236 '016035135'
2237 '035030016'
2238 '020016034'
2239 '035012012'
2240 '027776000'

2241 VE227(31)
2242 '002002007'                " 227
2243 '130135014'
2244 '030026031'
2245 '025016041'
2246 '135012033'
2247 '033012042'
2248 '135016025'
2249 '016026016'
2250 '027035135'
2251 '022027135'
2252 '016041031'
2253 '033016034'
2254 '034022016'
2255 '135040016'
2256 '020016027'
2257 '034135034'
2258 '031016014'
2259 '022017022'
2260 '014012035'
2261 '022016135'
2262 '012033022'
2263 '035021026'
2264 '016035022'
2265 '014135012'
2266 '033033012'
2267 '042135027'
```

```
'022016035'
'135035030'
'016020016'
'034035012'
'012027776'

UF228(12)
'002002010'                    " 228
'130135027'
'022016035'
'135013016'
'034035012'
'012027015'
'016135013'
'012027015'
'025016043'
'016033034'
'035033030'
'030026776'

UF229(13)
'002002011'                    " 229
'130135034'
'042027035'
'012014035'
'022034014'
'021016135'
'017030036'
'035135022'
'027135020'
'016035012'
'025025016'
'027013012'
'027015776'

UF230(27)
'002003000'                    " 230
'130135031'
'012033022'
'035016022'
'035034017'
'030036035'
'127135014'
'030015016'
'135030027'
'020016015'
'016017022'
'027022016'
'016033015'
'135030017'
'135040022'
'025015016'
'135030027'
'015016033'
'034035033'
'016031022'
'027020135'
'022027135'
'020016035'
'012025025'
```

```
2325 '016027013'
2326 '012027015'
2327 '776000000'

2328 VE231(13)
2329 '002003001'                " 231
2330 '130135027'
2331 '022016035'
2332 '135013016'
2333 '034035012'
2334 '012027015'
2335 '016135013'
2336 '012027015'
2337 '031030027'
2338 '034016033'
2339 '034035033'
2340 '030030026'
2341 '776000000'

2342 VE232(18)
2343 '002003002'                " 232
2344 '130135035'
2345 '016025016'
2346 '041014030'
2347 '015016135'
2348 '037030030'
2349 '033135013'
2350 '012027015'
2351 '031030027'
2352 '034016033'
2353 '135027022'
2354 '016035135'
2355 '020016022'
2356 '026031025'
2357 '016026016'
2358 '027035016'
2359 '016033015'
2360 '776000000'

2361 VE233(13)
2362 '002003003'                " 233
2363 '130135030'
2364 '031020016'
2365 '020016037'
2366 '016027135'
2367 '012012027'
2368 '035012025'
2369 '135031030'
2370 '027034016'
2371 '033034135'
2372 '035016135'
2373 '025012012'
2374         '020 776 000'
2375 VE234(11)
2376         '002 003 004'                  " 234
2377         '130 135 012'
2378         '027 030 027'
2379         '042 026 016'
2380         '135 040 016'
2381         '033 024 033'
```

```
2382          '036 022 026'
2383          '035 016 135'
2384          '035 016 135'
2385          '020 033 030'
2386          '030 035 776'

2387 UE235(13)
2388          '002 003 005'
2389          '130 135 030'
2390          '031 020 016'
2391          '020 016 037'
2392          '016 027 135'
2393          '012 012 027'
2394          '035 012 025'
2395          '135 031 025'
2396          '030 035 035'
2397          '016 033 034'
2398          '135 035 016'
2399          '135 025 012'
2400          '012 020 776'

2401 UINVGEC(22)
2402          A = 1
2403          S = 4
2404          SE18
2405          S = MA[- 6]
2406      U,  S - 13, Z
2407      N,  SUBCD(:RSE19)
2408          F = :MA[- 11]
2409          SUBCD(:RSE30)
2410          S = 18
2411          SE1
2412          F = 0
2413          MC = F
2414          DOS(MA[- 11])
2415          DO(ECV)
2416          MA[- 13] = F
2417          F = - MC[- 2]
2418          MA[- 15] = F
2419          F = MA[- 13]
2420          S = UGECPRICE
2421          GBILL + S, F
2422          SE115
2423          SE71

2424 QRSMUK2(10)
2425          CHEEN
2426          UCALCULATION
2427          S = UTRANSTIME, Z
2428      Y,  S = G
2429      Y,  UTRANSTIME = S
2430          S = - 0
2431          CGLSLOW = S
2432          S = 510
2433          CGLSTV[1] = S
2434          CTERUG
2435 CSTRINGSYM(14)
2436          A = 1
2437          S = 4
2438          SE18
```

```
2439          $ = MA[-6]
2440      U,  $ - 9 ,Z
2441      N,  SUBCD(:RSE19)
2442          F = :CGLSTV
2443          SUBCD(:UDNSS)
2444          G = A
2445          DO(MD[-1])
2446          $ = 11
2447          GBILL * S ,P
2448          SE115
2449          SE71
2450 CSETSTRING(16)
2451          A = 1
2452          $ = 4
2453          SE18
2454          $ = MA[-6]
2455      U,  $ - 13 ,Z
2456      N,  SUBCD(:RSE19)
2457          F = :MA[-11]
2458          SUBCD(:RSE29)          "CHECK FORMAL STRING
2459          B = :CGLSTV
2460          G = MA[-9]
2461          SUBCD(:UINSTV)
2462          B = MA
2463          $ = 10
2464          GBILL * S ,P
2465          SE115
2466          SE71
```

two procedures
that operate on (static) strings

```
 0 CORRECTION(177)
 1          G = MT[25]                    " CINVCOR
 2          SUBCD(:UINSTV)
 3          F = - 9                       " BUILDING UP WORD
 4          MC = F                        " AND COUNTER ON STACK
 5          G = UWHO
 6          F - 1
 7          MC = G                        " INVARIANT STARTING ADDRESS PRELIMINARY IDENTIFICATION
 8          JUMP(2)
 9          CHEEN
10          UFILLSTRING
11          A = :M0[0]
12          CHEEN
13          USP
14     U,   S - 9, P                      " ¬ DIGIT ?
15     N,   A = 1
16     N,   M[B - 2] + A, Z
17     N,   JUMP( - 9)
18          M[B - 2] = S                  " SAVE FIRST NONDIGIT TEMPORARILY
19          F = :MC[ - 6]
20          SUBCD(:UDNSS)
21     U,   A - 510, Z                    " END OF STRING ?
22          S = A
23     Y,   JUMP(4)
24          CHEEN
25          UFILLSTRING
26          JUMP( - 8)
27          CINVCOR
28          CHEEN
29          UFILLSTRING
30          S = M[B - 2]                  " RETRIEVE NONDIGIT
31     U,   S - 63, P                     " ¬ (LETTER OR DIGIT)
32     Y,   JUMP(4)
33          A = :M0[0]
34          CHEEN
35          USP
36          JUMP( - 6)
37          S - 119, Z
38     Y,   JUMP(2)
39          CHEEN
40          UNINT
41          B - 6
42          S = 2
43          UKIND = S
44          UDEAF
45          SUBCD(:UCLAIMCONRED)
46          UTREXR
47          S = - USPECIALPM, P
48     Y,   S = URUNACCERT, Z
49          A = 1
50          G = UWHO
51     N,   JUMP(7)
52          UTREXV
53          F = - 0
54          MC = F
55          B + 3
56          SE33
57          UNAC
58          UINVMBEGIN
59          UPROGID = G
```

```
60          UKILLVAR = A, E
61      Y,  UCLAIMRUN = A
62          S = UCURRUNNUM
63          URUNNUM = S
64          UCURRUNNUM + A
65          UTREXV
66          S = :M0[0]
67          SUBCD(:URROGNAME)
68          G = MT[-11]
69          SUBCD(:CRRINTSTRING)
70          SUBCD(:UCONFREEWHITE)
71          DO(MD[-1])

72          S = 19
73          SE0
74          S = 30
75          SE1
76          S = - 1
77          MA[3] = S
78          S = 0
79          MA[5] = S
80          B + 2
81          DOS(RSE76)
82          0
83          2
84          S = 13
85          SE73
86          CINIRATA
87          MA[12] = B
88          MA[14] = - 8
89          MA[11] = - 8
90          B + 2
91          DOS(RSE76)
92          0
93          1
94          S = 13
95          SE73
96          CSYMINSR
97          E - 1, Z
98      Y,  JUMR( - 10)
99          E - 1, Z
100     Y,  JUMR(70)
101         CHEEN
102         CNUMCOR
103         MA[6] = S
104         CHEEN
105         CNUMCOR
106         MA[7] = S
107     U,  S + 1, R
108         S - 2, E
109     N,  JUMR(3)
110         S = 4
111         SE33
112         CORRERR

113         CHEEN
114         CNUMCOR
115         MA[8] = S
116         B + 2
117         S = 9
```

This is a tape editing program

" LINENUM := 0

" :READCORR

" END OF CODE ?

" END OF TAPE ?
" GOTO AFMAKEN

" CORLINE

" TYPE

" ¬ 0≤ TYPE ≤ 2 ?

" N

```
118          SE73
119          CHAREI
120          E - 119, Z                    " NLCR ?
121      N,  $ = 6
122      N,  JUMP( - 12)
123          $ = MA[5]                     " LINENUM
124      U,  $ - MA[6], P                  " > CORLINE
125      Y,  $ = 5
126      Y,  JUMP( - 16)
127      U,  $ - MA[6], Z                  " : COPY
128      Y,  JUMP(5)
129          CHEEN
130          CRUNCHLINE
131          $ = 1
132          PLUS$(MA[5])                  " LINENUM := LINENUM + 1
133          JUMP( - 7)                    " GOTO COPY
134          $ = MA[7]                     " :NEXTLINE
135      U,  $ - 1, Z
136      Y,  JUMP(25)
137          MA[13] = B                    " SKIPNLCR := TRUE
138          MA[11] = - B                  " CORR := FALSE
139          B + 2                         " :L
140          $ = 9
141          SE73
142          CHAREI2
143          MA[2] = G, P                  " IF (CHAR := RECHAR) > 0
144      Y,  JUMP(11)
145          E + 2, Z                      " CHAR = - 2
146      Y,  $ = 2
147      Y,  JUMP( - 37)
148          CHEEN
149          CODEDEF
150          DO(MD[ - 1])
151      Y,  JUMP( - 13)                   " IFCODEDEF THEN GOTO L
152          $ + 1, Z                      " END OF  TAPE ?
153      Y,  $ = 9
154      N,  $ = 10
155          JUMP( - 45)
156          E - 119, Z
157      N,  MA[13] = - B                  " SKIPNLCR := FALSE
158      Y,  $ = - MA[13], P               " IF - SKIPNLCR
159      N,  JUMP( - 21)                   " GOTO L
160          $ = 1
161          PLUS$(MA[5])                  " LINENUM := LINENUM + 1
162          $ = MA[7], Z
163      Y,  JUMP(3)
164          MA[11] = B                    " CORR := TRUE
165          CHEEN
166          CRUNCHLINE
167          $ = 1
168          MA[8] - $, P                  " IF ( N := N - 1) > 0
169      Y,  JUMP( - 36)                   " THEN GOTO NEXTLINE
170          JUMP( - 82)                   " GOTO READCORR
171          MA[14] = B                    " AFM := TRUE
172          CHEEN                         " :AFMAKEN
173          CRUNCHLINE
174          JUMP( - 3)
175          $ = 0
176          SE33
177          UNORMALEXIT
```

```
178 CORRERR(43)
179      U,  $ = MA[11], R                    " CORR = IBUE ?
180      Y,  $ + 10
181          MC = $
182          B + 2
183          $ = 9
184          SE73
185          CRUNOUT
186          $ = MC[ - 1]
187          A = 2
188          CHEEN
189          CRUVISNUM
190          $ = :MG[4]
191          MC = $                           " SAVE POINTER
192          A = M$                           " GET SYMBOL
193      U,  A - 9, R
194      Y,  A = 36                           " FOR BLANK SPACE
195          G = MT[17]                       " CVISRU
196          LUA(2)
197          G + A
198          SUBCD(:UINSTV)
199          F = :MC[ - 3]
200          SUBCD(:UDNSS)
201      U,  A - 510, Z
202      Y,  JUMP(11)
203          MC = A
204          DO(MD[ - 1])
205          $ = :MC[ - 1]
206          B + 2
207          SUBC(:RSE78)
208          $ = 13
209          SE73
210          CRUOCT1
211          B - 1
212          JUMP( - 14)
213          CVISRU
214          B - 3
215          $ = 1
216          $ + MC[ - 1]
217      U,  $ - :MC[ - 2], R                 " TRALING SPACE WILL NOT BE PUNCHED
218      N,  JUMP( - 28)
219          $ = 0
220          SE33
221          UNORMALEXIT

222 CNUMCOR(20)
223          $ = 0
224          MC = $
225          B + 2
226          $ = 9
227          SE73
228          CHARE1
229          F + 0, R
230          F - 9, E
231          $ = MC[ - 1]
232      Y,  JUMP(7)
233          TENS
234          $ + :MG[9]
235      U,  $ - 32 767, R                    " NUMBER TOO LARGE ?
```

```
236      N,  JUMP( - 13)
237          $ = 8
238          SE33
239          CORRERR
240          F - 78, Z                    " KOMMA ?
241      Y,  SUBCD(:CTERUGADRES)
242          JUMP( - 6)

243  CRUNCHLINE(105)
244          $ = - 0
245          MA[9] = S                    " NLCRS := 0
246          MA[15] = - S                 " START := IBUF
247          B + 2                        " :NEXTCHAR
248          $ = 9
249      U,  S = MA[11], R
250      Y,  JUMP(3)
251          SE73
252          CHARE2
253          JUMP(2)
254          SE73
255          CHARE1
256          MA[2] = G, R
257      Y,  JUMP(20)
258          F + 1, Z                     " END OF CODE ?
259      Y,  JUMP(3)
260          $ = 1
261          SE33
262          CORRERR
263          CHEEN
264          CODEDEF
265          DO(MD[ - 1])
266      Y,  JUMP( - 20)
267      U,  $ + 1, Z                     " END OF TAPE ?
268      N,  $ = 10
269      N,  JUMP( - 9)
270          $ = MA[15], R                " START ?
271      Y,  $ = -MA[11], R               " ^ ¬ CORR ?
272      Y,  $ = MA[14], R                " ^ AFM ?
273      N,  $ = 9
274      N,  JUMP( - 14)
275          $ = 0
276          SE33
277          UNORMALEXIT
278      U,  $ = MA[15], R                " START ?
279      N,  JUMP(50)
280          $ = MA[2]
281      U,  $ - 119, Z
282      Y,  $ = 1
283      Y,  MA[9] + S
284      Y,  JUMP( - 38)
285          $ = MA[9], R                 " NLCRS > 0
286      N,  JUMP(26)
287      U,  $ = MA[12], R                " BEGIN ?
288      Y,  JUMP(11)
289          B + 2
290          DOS(RSE76)
291          0
292          0
293          $ = 13
294          SE73
```

```
295          CRUOCT1
296          B + 2
297          S = 9
298          $E73
299          CRUNOUT
300      U,  $ = ULIST, R
301      N,  JUMP(4)
302          B + 2
303          S = 9
304          $E73
305          CNLCR
306          B + 2
307          S = 9
308          $E73
309          CRUNLCR1
310          S = 1
311          MIN$(MA[9]), R
312      Y,  JUMP( - 13)
313          MA[15] = S                  " START := FALSE
314          S = 1
315          MA[3] + S                   " NEWLINENUM + 1
316      U,  $ = ULIST, R
317      N,  JUMP(20)
318          B + 2
319          DOS(RSE76)
320          0
321          4
322          DOS(RSE76)
323          0
324          0
325          S = :MA[3]
326          SUBC(:RSE78)
327          S = 21
328          $E73
329          CABSEIXT
330      U,  $ = ULIST, R
331      N,  JUMP(6)
332          B + 2
333          S = :MA[2]
334          SUBC(:RSE78)
335          S = 13
336          $E73
337          CRRCHAR
338          B + 2
339          S = :MA[2]
340          SUBC(:RSE78)
341          S = 13
342          $E73
343          CRUCHARE1
344          S = MA[2]
345          S - 119, Z
346      N,  JUMP( - 100)
347          MA[12] = S                  " BEGIN := FALSE
348          CTERUG
349          CINVCOR(5)
350          '135 014 030'               "  CO
351          '033 033 016'               " RRE
352          '014 035 022'               " CTI
353          '030 027 135'               " ON
354          '776 000 000'
```

```
355 UINVPLOTTEXT(319)
356         A = 1
357         $ = 6
358         $E18
359         $ = MA[- 6]
360         $ - 41, Z
361     N,  SUBCD(:RSE19)
362         F = :RTX[UA1]
363         SUBCD(:RSE20)
364         F = :RTY[UA1]
365         SUBCD(:RSE20)
366         F = :RTANGLE[UA1]
367         SUBCD(:RSE20)
368         F = :RTHEIGHT[UA1]
369         SUBCD(:RSE20)
370         F = :RTITALICITY[UA1]
371         SUBCD(:RSE20)
372         F = :RTERST[UA1]
373         SUBCD(:RSE21)
374         F = :RTI[UA1]
375         SUBCD(:RSE20)
376         F = :RTTEXT[UA1]
377         SUBCD(:RSE29)
378         $ = 16
379         $E1
380         DOS(RTX[UA1])
381         DO(ERV)
382         RTX[UA1] = F
383         DOS(RTY[UA1])
384         DO(ERV)
385         RTY[UA1] = F
386         DOS(RTERST[UA1])
387         DO(ERV)
388         RTERST[UA1] = G
389         DOS(RTI[UA1])
390         DO(ERV)
391         $ = F, Z
392     N,  SUBCD(:RSE88)
393         RTI[UA1] = G
394         G = UPLSTREAM$
395         G = MG[0]
396         UBLOCKSTR = G
397         $ = MG[6], P
398     Y,  JUMP(2)
399         CHEEN
400         UINVBEGINPLDOC

401         $ = 35
402         $E0
403         $ = 29
404         $E1
405         G = RTTEXT[2]
406         SUBCD(:UINSTV)
407         F = - 1
408         RTJ[UA2] = G
409         G = RTI, P
410     N,  F = - F
411         RTABS[UA2] = G
412         F - 1, P
```

more plotter procedures

this constant transforms M1 addressing in MA addressing

apparently the ruling blockheight here is 1.
(We are in the "fictious" parameterblock of this procedure)

```
413      N,  JUMP(5)
414          F - 11, F
415      Y,  JUMP(100)                    " GOTO MISTAKE
416          F + 142
417          RTABSI[UA2] = G              " a local of this procedure body
418          JUMP(14)                     " GOTO SYMBOL
419          G = RTJ[UA2]                 " STRING
420          F + 1
421          RTJ[UA2] = G
422          F = :MC[- 3]
423          S = MG[0]
424      U,  S - UCON2, P
425      N,  A = MS[0]
426      N,  S + 1
427      N,  MG[0] = S
428      Y,  SUBCD(:UDNSS)
429          RTABSI = A
430      U,  A - 510, Z
431          DO(MD[- 1])
432      Y,  JUMP(236)                    " GOTO END
433          DOS(RTHEIGHT)                " SYMBOL
434          DO(ERV)
435          F/7
436          RTHE[UA2] = F, P
437      N,  F = - F
438          RTABSHE[UA2] = F
439          DOS(RTANGLE)
440          DO(ERV)
441          F * MT[6]
442          RTAN[UA2] = F
443          DOS(RTITALICITY)

444          DO(ERV)
445          F * MT[2]
446          RTIT[UA2] = F
447          JUMP(2)
448          - 1498177
449          + 37032595
450          F = RTAN[UA2]
451          SUBC(:UCOS)
452          F * RTABSHE[UA2]
453          RTHECOSAN[UA2] = F
454          F = RTAN[UA2]
455          SUBC(:USIN)
456          F * RTABSHE[UA2]
457          RTHESINAN[UA2] = F
458          F = RTIT[UA2]
459          SUBC(:UCOS)
460          RTCOSIT[UA2] = F
461          F = RTIT[UA2]
462          SUBC(:USIN)
463          F/RTCOSIT[UA2]
464          RTTANIT[UA2] = F
465          F = RTCOSIT[UA2], Z
466      N,  F = 1, E
467      N,  F = - 1
468          RTCOSIT[UA2] = F
469  G = RTJ[UA2]
470          F - 0, P
471      Y,  JUMP(86)
```

now M2 addressing → MA since we know that A points to the local ref. point of the current block.

PTABSI is made now to the assembler as a M2[ ] address (see table)

```
472         G = RTEIRST, R
473     N,  JUMP(40)
474         RTEIRST = - 8
475         RTSHIFT = - 8
476         USTATE64 = 8
477         E = - RTHE[UA2]
478         E - 0, R
479     N,  JUMP(5)
480         E = RTX
481         RTX0 = E

482         E = RTY
483         RTY0 = E
484         JUMP(34)
485         E = UCLEAR
486         MC = E
487         S = :RTX
488         SUBCD(:RSE77)
489         DOS(RSE76)
490         + 0
491         + 0
492         DOS(RSE76)
493         + 0
494         + 5
495         S = 21
496         SE73
497         UINVPLOT
498         RTX0 = E
499         E = UCLEAR
500         MC = E
501         DOS(RSE76)
502         + 0
503         + 0
504         S = :RTY
505         SUBCD(:RSE77)
506         DOS(RSE76)
507         + 0
508         + 6
509         S = 21
510         SE73
511         UINVPLOT
512         RTY0 = E
513         JUMP(5)
514         G = - USTATE64, R
515     N,  JUMP(3)
516         A = 243                          " MISTAKE
517         SUBCD(:UDYNERR)
518         JUMP(150)                        " GOTO END
519         G = - PTI

520         E - 0, R
521     Y,  S = - RTSHIFT, R
522     N,  JUMP(16)
523         E = RTHECOSAN[UA2]
524         E - RTHESINAN[UA2]
525         E + E
526         E - RTHESINAN[UA2]
527         E * RTCOSIT[UA2]
528         E - RTX0
529         RTX0 = - E
```

```
530          F = RTHECOSAN[UA2]
531          F + RTHESINAN[UA2]
532          F + F
533          F + RTHECOSAN[UA2]
534          F * RTCOSIT[UA2]
535          F - RTY0
536          RTY0 = - F
537          RTSHIFT = B
538          JUMP(19)
539          G = - RTI
540          F - 0, P
541      N,  S = - RTSHIFT, P
542      Y,  JUMP(15)
543          F = RTHECOSAN[UA2]
544          F - RTHESINAN[UA2]
545          F + F
546          F - RTHESINAN[UA2]
547          F * RTCOSIT[UA2]
548          F + RTX0
549          RTX0 = F
550          F = RTHECOSAN[UA2]
551          F + RTHESINAN[UA2]
552          F + F
553          F + RTHECOSAN[UA2]
554          F * RTCOSIT[UA2]
555          F + RTY0
556          RTY0 = F
557          RTSHIFT = - B
558          F = - 2
559          RTIREN[UA2] = G
560          F = RTX0
561          RTLX[UA2] = F
562          F = RTY0
563          RTLY[UA2] = F
564          A = RTABS[UA2]
565          A + MT[108], Z
566          SUBCD(:RSE39)
567          SUBCD(:UINSTV)
568          F = :MC[- 3]
569          SUBCD(:UDNSS)
570      U,  A - 510, Z
571      N,  S = A
572      N,  RUA(3)
573      N,  RTXKI = A
574          DO(MD[- 1])
575      Y,  B - 3
576      Y,  JUMP(80)
577          S '*' 7
578          G = S
579          F * RTCOSIT[UA2]
580          RTYK[UA2] = F
581          G = - RTXK[UA2]
582          F + 5, P
583      N,  JUMP(37)
584          F = RTYK[UA2]
585          F * RTTANIT[UA2]
586          G + RTXK[UA2]
587          RTXK[UA2] = F
588          F = UCLEAR
```

```
589          MC = F
590          DOS(R82E3)
591          + 130
592          UINVPLOTTEXT[246]
593          DOS(R82E3)
594          + 130
595          UINVPLOTTEXT[254]

596          S = :RTIREN[UA2]
597          SUBC(:RSE78)
598          S = 21
599          SE73
600          UINVPLOT
601          JUMP(16)
602          F = RTXK
603          F * RTHECOSAN
604          F + RTLX
605          MC = F
606          F = - RTYK
607          F * RTHESINAN
608          F + MC[- 2]
609          SUBCD(:RSE72)
610          F = RTYK
611          F * RTHECOSAN
612          F + RTLY
613          MC = F
614          F = RTXK
615          F * RTHESINAN
616          F + MC[- 2]
617          SUBCD(:RSE72)
618          F = - 1
619          RTIREN[UA2] = G
620          JUMP(32)
621          G = RTXK[UA2]
622          F - 7, Z
623      Y,  S = - 2
624      Y,  RTIREN[UA2] = S
625      Y,  JUMP(27)
626          F + 1, Z
627      N,  JUMP(13)
628          F = - RTTANIT[UA2]
629          F * RTHECOSAN[UA2]
630          F + RTHESINAN[UA2]
631          F * RTYK[UA2]
632          F - RTLX[UA2]
633          RTLX[UA2] = - F
634          F = RTTANIT[UA2]

635          F * RTHESINAN[UA2]
636          F + RTHECOSAN[UA2]
637          F * RTYK[UA2]
638          F + RTLY[UA2]
639          RTLY[UA2] = F
640          JUMP(12)
641          F = - RTTANIT[UA2]
642          F * RTHECOSAN[UA2]
643          F + RTHESINAN[UA2]
644          F * RTYK[UA2]
645          F + RTLX[UA2]
646          RTLX[UA2] = F
```

```
647         F = RTTANIT[UA2]
648         F * RTHESINAN[UA2]
649         F + RTHECOSAN[UA2]
650         F * RTYK[UA2]
651         F - RTLY[UA2]
652         RTLY[UA2] = - F
653         S = RTPLOTTEXTPRICE1
654         GBILL + S, P
655         SE115
656         JUMP(- 89)
657         F = 6
658         F * RTHECOSAN[UA2]
659         F + RTX0
660         RTX0 = F
661         F = 6
662         F * RTHESINAN[UA2]
663         F + RTY0
664         RTY0 = F
665         S = RTPLOTTEXTPRICE2
666         GBILL + S, P
667         SE115
668         JUMP(- 250)
669         S = RTPLOTTEXTPRICE                    " END
670         GBILL + S, P
671         SE115
672         SE6
673         SE71
674         UINVENTRANCE

675 UINVENTRANCE(143)
676         UR0
677         UR1
678         UR2
679         UR3
680         UR4
681         UR5
682         UR6
683         UR7
684         UR8
685         UR9
686         URA
687         URB
688         URC
689         URD
690         URE
691         URF
692         URG
693         URH
694         URI
695         URJ
696         URK
697         URL
698         URM
699         URN
700         URO
701         URP
702         URQ
703         URR
704         URS
705         URT
```

conversion table for plotter characters

```
706         URU
707         URV
708         URW
709         URX
710         URY
711         URZ
712         URBLANK
713         URA
714         URB
715         URC
716         URD
717         URE
718         URF
719         URG
720         URHM
721         URI
722         URJ
723         URK
724         URL
725         URM
726         URN
727         URO
728         URP
729         URQ
730         URR
731         URS
732         URT
733         URU
734         URV
735         URW
736         URX
737         URY
738         URZ
739         URBLANK
740         URPLUS
741         URMINUS
742         URTIMES
743         URDIV
744         URINTDIV
745         URHUR
746         UREQUALS
747         URUNEQUALS
748         URSMALLER
749         URATMOST
750         URGREATER
751         URATLEAST
752         URNOT
753         UREQUIVALENT
754         URIMPLIES
755         UROR
756         URAND
757         URBLANK
758         URBLANK
759         URBLANK
760         URBLANK
761         URBLANK
762         URBLANK
763         URCOMMA
764         URPERIOD
765         URDOLLARSIGN
```

```
766 URCOLON
767           URSEMICOLON
768           URBLANK
769           URBLANK
770           URBLANK
771           URBLANK
772           URBLANK
773           URBLANK
774           UROPEN
775           URCLOSE
776           URSUB
777           URBUS
778           URSTRINGOPEN
779           URSTRINGCLOSE
780           URMARK2
781           URMARK3
782           URMARK4
783           URMARK7
784           URMARK6
785           URMARK5
786           URMARK11
787           URMARK8
788           URMARK10
789           URMARK9
790           URMARK12
791           URBLANK
792           URBLANK
793           URBLANK
794           URBLANK
795           URBLANK
796           URAROS
797           URDITOS
798           URQUESTIONMARK
799           URBLANK
800           URBLANK
801           URBLANK
802           URUNDER
803           URSTROKE
804           URBLANK
805           URBLANK
806           URBLANK
807           URBLANK
808           URMARK2
809           URMARK3
810           URMARK4
811           URMARK5
812           URMARK6
813           URMARK7
814           URMARK8
815           URMARK9
816           URMARK10
817           URMARK11
818           URMARK12

819 UR0(4)
820           '010 001 006'
821           '017 037 046'
822           '041 030 010'
823           '776 000 000'
824 UR1(2)
```

```
825           '020 027 016'
826           '776 000 000'
827 UR2(4)
828           '005 006 017'
829           '037 046 044'
830           '001 000 040'
831           '776 000 000'
832 UR3(5)
833           '001 010 030'
834           '041 043 034'
835           '014 034 045'
836           '046 037 017'
837           '006 776 000'
838 UR4(2)
839           '030 037 003'
840           '043 776 000'
841 UR5(4)
842           '001 010 030'
843           '041 043 034'
844           '004 007 047'
845           '776 000 000'
846 UR6(5)
847           '003 014 034'
848           '043 041 030'
849           '010 001 006'
850           '017 037 046'
851           '776 000 000'
852 UR7(2)
853           '006 007 047'
854           '010 776 000'
855 UR8(6)
856           '014 003 001'
857           '010 030 041'
858           '043 034 014'
859           '005 006 017'
860           '037 046 045'
861           '034 776 000'
862 UR9(5)
863           '001 010 030'
864           '041 046 037'
865           '017 006 004'
866           '013 033 044'
867           '776 000 000'
868 URA(4)
869           '043 034 014'
870           '003 001 010'
871           '030 041 040'
872           '044 776 000'
873 URB(4)
874           '007 000 001'
875           '010 030 041'
876           '043 034 014'
877           '003 776 000'
878 URC(3)
879           '041 030 010'
880           '001 003 014'
881           '034 043 776'
882 URD(4)
883           '043 034 014'
```

```
884             '003 001 010'
885             '030 041 040'
886             '047 776 000'
887 URE(4)
888             '041 030 010'
889             '001 003 014'
890             '034 043 042'
891             '002 776 000'

892 URF(3)
893             '000 004 024'
894             '004 006 017'
895             '037 046 776'
896 URG(5)
897             '053 001 010'
898             '030 041 046'
899             '037 017 006'
900             '004 013 033'
901             '044 776 000'
902 URH(3)
903             '007 000 003'
904             '014 034 043'
905             '040 776 000'
906 URI(2)
907             '010 030 020'
908             '024 014 776'
909 URJ(3)
910             '053 001 010'
911             '030 041 047'
912             '776 000 000'
913 URK(3)
914             '017 010 011'
915             '044 022 040'
916             '776 000 000'
917 URL(2)
918             '010 030 020'
919             '027 017 776'
920 URM(4)
921             '000 004 003'
922             '014 023 020'
923             '023 034 043'
924             '040 776 000'
925 URN(3)
926             '000 004 003'
927             '014 034 043'
928             '040 776 000'

929 URO(4)
930             '001 010 030'
931             '041 043 034'
932             '014 003 001'
933             '776 000 000'
934 URP(4)
935             '053 000 007'
936             '006 017 037'
937             '046 044 033'
938             '013 004 776'
939 URQ(4)
940             '053 040 047'
941             '046 037 017'
```

internal representation of the moves necessary to draw these characters on the plotter

```
942           '006 004 013'
943           '033 044 776'
944 URR(3)
945           '000 004 003'
946           '014 034 043'
947           '776 000 000'
948 URS(4)
949           '001 010 030'
950           '041 032 012'
951           '003 014 034'
952           '043 776 000'
953 URT(4)
954           '017 014 004'
955           '024 014 011'
956           '020 030 041'
957           '776 000 000'
958 URU(3)
959           '004 001 010'
960           '030 041 040'
961           '044 776 000'
962 URV(2)
963           '004 020 044'
964           '776 000 000'

965 URW(4)
966           '004 001 010'
967           '021 023 021'
968           '030 041 044'
969           '776 000 000'
970 URX(2)
971           '000 044 022'
972           '004 040 776'
973 URY(4)
974           '053 001 010'
975           '030 041 047'
976           '044 033 013'
977           '004 007 776'
978 URZ(2)
979           '004 044 000'
980           '040 776 000'
981 URA(4)
982           '000 004 044'
983           '046 037 017'
984           '006 004 044'
985           '040 776 000'
986 URB(5)
987           '000 007 037'
988           '046 045 034'
989           '004 034 043'
990           '041 030 000'
991           '776 000 000'
992 URC(4)
993           '045 046 037'
994           '017 006 001'
995           '010 030 041'
996           '042 776 000'
997 URD(3)
998           '000 007 037'
999           '046 041 030'
1000          '000 776 000'
```

```
1001 URE(3)
1002          '047 007 004'
1003          '034 004 000'
1004          '040 776 000'
1005 URF(3)
1006          '047 007 004'
1007          '034 004 000'
1008          '776 000 000'
1009 URG(4)
1010          '045 046 037'
1011          '017 006 001'
1012          '010 030 041'
1013          '043 033 776'
1014 URHH(3)
1015          '000 007 004'
1016          '044 047 040'
1017          '776 000 000'
1018 URI(3)
1019          '010 030 020'
1020          '027 017 037'
1021          '776 000 000'
1022 URJ(2)
1023          '001 010 030'
1024          '041 047 776'
1025 URK(3)
1026          '007 000 003'
1027          '047 014 040'
1028          '776 000 000'
1029 URL(2)
1030          '007 000 040'
1031          '776 000 000'

1032 URM(2)
1033          '000 007 024'
1034          '047 040 776'
1035 URN(2)
1036          '000 007 040'
1037          '047 776 000'
1038 URO(2)
1039          '007 000 040'
1040          '047 007 776'
1041 URR(3)
1042          '000 007 037'
1043          '046 045 034'
1044          '004 776 000'
1045 URQ(5)
1046          '010 001 006'
1047          '017 037 046'
1048          '041 030 010'
1049          '070 022 040'
1050          '776 000 000'
1051 URR(4)
1052          '000 007 037'
1053          '046 045 034'
1054          '004 034 043'
1055          '040 776 000'
1056 URS(5)
1057          '001 010 030'
1058          '041 043 034'
1059          '014 005 006'
```

```
1060          '017 037 046'
1061          '776 000 000'
1062 URT(2)
1063          '020 027 007'
1064          '047 776 000'

1065 URU(3)
1066          '007 001 010'
1067          '030 041 047'
1068          '776 000 000'
1069 URV(2)
1070          '007 020 047'
1071          '776 000 000'
1072 URW(2)
1073          '007 010 024'
1074          '030 047 776'
1075 URX(3)
1076          '000 024 007'
1077          '024 047 024'
1078          '040 776 000'
1079 URY(2)
1080          '020 024 007'
1081          '024 047 776'
1082 URZ(2)
1083          '007 047 000'
1084          '040 776 000'
1085 URPLUS$(2)
1086          '003 043 023'
1087          '025 021 776'
1088 URMINUS(1)
1089          '003 043 776'
1090 URTIMES(2)
1091          '001 045 023'
1092          '005 041 776'
1093 URDIV(1)
1094          '000 047 776'
1095 URINTDIV(5)
1096          '012 010 030'
1097          '032 012 070'
1098          '014 016 036'
1099          '034 014 070'
1100          '003 043 776'

1101 URNUP(2)
1102          '020 026 004'
1103          '026 044 776'
1104 UREQUALS(2)
1105          '002 042 070'
1106          '044 004 776'
1107 URUNEQUALS(3)
1108          '002 042 070'
1109          '044 004 070'
1110          '046 000 776'
1111 URSMALLER(2)
1112          '041 003 045'
1113          '776 000 000'
1114 URATMOST(3)
1115          '000 040 070'
1116          '041 003 045'
1117          '776 000 000'
```

```
1118 URGREATER(2)
1119          '005 043 001'
1120          '776 000 000'
1121 URATLEAST(3)
1122          '005 043 001'
1123          '070 000 040'
1124          '776 000 000'
1125 URNON(2)
1126          '004 044 042'
1127          '776 000 000'
1128 UREQUIVALENT(3)
1129          '001 041 070'
1130          '043 003 070'
1131          '005 045 776'
1132 URIMPLIES(3)
1133          '001 031 042'
1134          '044 035 005'
1135          '776 000 000'

1136 URDR(2)
1137          '006 020 046'
1138          '776 000 000'
1139 URAND(2)
1140          '000 026 040'
1141          '776 000 000'
1142 URCOMMA(3)
1143          '052 010 032'
1144          '034 014 012'
1145          '022 010 776'
1146 URPERIOD(2)
1147          '012 010 030'
1148          '032 012 776'
1149 URDRORREDTEN(4)
1150          '052 002 013'
1151          '010 070 020'
1152          '040 043 023'
1153          '020 776 000'
1154 URCOLON(4)
1155          '012 010 030'
1156          '032 012 070'
1157          '014 016 036'
1158          '034 014 776'
1159 URSEMICOLON(5)
1160          '052 010 032'
1161          '034 014 012'
1162          '022 010 070'
1163          '015 017 037'
1164          '035 015 776'
1165 URDREN(2)
1166          '040 031 036'
1167          '047 776 000'
1168 URCLOSE(2)
1169          '000 011 016'
1170          '007 776 000'

1171 URSUB(2)
1172          '040 020 027'
1173          '047 776 000'
1174 URBUS(2)
1175          '000 020 027'
```

```
1176          '007 776 000'
1177 VRSTRINGOREN(3)
1178          '062 023 025'
1179          '047 035 045'
1180          '043 023 776'
1181 VRSTRINGCLOSE(3)
1182          '062 003 015'
1183          '005 007 027'
1184          '025 003 776'
1185 VRAPOS(3)
1186          '062 015 017'
1187          '027 025 015'
1188          '776 000 000'
1189 VRDITOS(5)
1190          '062 015 005'
1191          '007 017 015'
1192          '070 025 035'
1193          '037 027 025'
1194          '776 000 000'
1195 VRQUESTIONMARK(5)
1196          '006 007 047'
1197          '046 013 012'
1198          '032 070 031'
1199          '011 010 030'
1200          '031 776 000'
1201 VRUNDER(2)
1202          '052 000 040'
1203          '776 000 000'
1204 VRSTROKE(1)
1205          '020 027 776'
1206 VRMARK2(3)
1207          '003 043 023'
1208          '025 021 023'
1209          '776 000 000'

1210 VRMARK3(3)
1211          '001 045 023'
1212          '005 041 023'
1213          '776 000 000'
1214 VRMARK4(4)
1215          '001 045 023'
1216          '005 041 023'
1217          '003 043 023'
1218          '776 000 000'
1219 VRMARK5(5)
1220          '001 045 023'
1221          '005 041 023'
1222          '003 043 023'
1223          '025 021 023'
1224          '776 000 000'
1225 VRMARK6(3)
1226          '021 023 005'
1227          '023 045 023'
1228          '776 000 000'
1229 VRMARK7(3)
1230          '001 041 005'
1231          '045 001 023'
1232          '776 000 000'
1233 VRMARK8(2)
1234          '041 005 045'
```

```
1235          '001 023 776'
1236  URMARK9(3)
1237          '021 001 005'
1238          '045 041 021'
1239          '023 776 000'
1240  URMARK10(3)
1241          '021 003 025'
1242          '043 021 023'
1243          '776 000 000'

1244  URMARK11(4)
1245          '062 005 014'
1246          '024 035 036'
1247          '027 017 006'
1248          '005 776 000'
1249  URMARK12(3)
1250          '010 014 004'
1251          '044 034 031'
1252          '040 776 000'
1253  URBLANK(1)
1254          '776 776 776'

1255  VINVEIXPLOT(1)
1256          JUMP(1)
1257  VINVABSEIXPLOT(1)
1258          JUMP(0)
1259  VINVELORPLOT(121)
1260          A = 1
1261          S = 5
1262          SE18
1263          $ = MA[- 6]
1264          $ - 49, Z
1265      N,  SUBCD(:RSE19)
1266          F = :RNX[UA1]
1267          SUBCD(:RSE20)
1268          F = :RNY[UA1]
1269          SUBCD(:RSE20)
1270          F = :RNANGLE[UA1]
1271          SUBCD(:RSE20)
1272          F = :RNHEIGHT[UA1]
1273          SUBCD(:RSE20)
1274          F = :RNITALICITY[UA1]
1275          SUBCD(:RSE20)
1276          F = :RNFIRST[UA1]
1277          SUBCD(:RSE21)
1278          F = :RNI[UA1]
1279          SUBCD(:RSE20)
1280          F = :RNN[UA1]
1281          SUBCD(:RSE20)
1282          F = :RNM[UA1]
1283          SUBCD(:RSE20)
1284          F = :RNNUMBER[UA1]
1285          SUBCD(:RSE20)
1286          $ = 75
1287          SE1
1288          DOS(RNX[UA1])
1289          DO(ERV)
1290          RNX[UA1] = F
1291          DOS(RNY[UA1])
1292          DO(ERV)
```

```
1293          RNY[UA1] = F
1294          DOS(RNFIRST[UA1])
1295          DO(ERV)
1296          RNFIRST[UA1] = G
1297          DOS(RNI[UA1])
1298          DO(ERV)
1299          $ = F, Z
1300      N,  SUBCD(:RSE88)
1301          RNI[UA1] = G
1302          DOS(RNN[UA1])
1303          DO(ERV)
1304          $ = F, Z
1305      N,  SUBCD(:RSE88)
1306          RNN[UA1] = G
1307          DOS(RNM[UA1])
1308          DO(ERV)
1309          $ = F, Z
1310      N,  SUBCD(:RSE88)
1311          RNM[UA1] = G
1312          DOS(RNNUMBER[UA1])
1313          DO(ERV)
1314          RNNUMBER[UA1] = F
1315          G = URLSTREAM$
1316          G = MG[0]
1317          UBLOCKSTR = G
1318          $ = MG[6], F
1319      Y,  JUMP(2)
1320          CHEEN
1321          UINVBEGINRLDOC
1322          $ = MA[- 5]
1323          $ - MT[56], Z
1324      N,  $ = $, F
1325      N,  $ = 0
1326          MC = $
1327          G = RNN[UA1]
1328          MC = G
1329          G = RNM[UA1]
1330          MC = G
1331          F = RNNUMBER[UA1]
1332          MC = F
1333          CHEEN
1334          CONVERTNUM

1335          $ = 510
1336          MC = $
1337          F + 4
1338          $ = 2080
1339          RNNUMBER[UA1 + 1] = $
1340          RNNUMBER[UA1 + 2] = G
1341          $ = MG[0]
1342      U,  $ - 510, Z
1343      Y,  JUMP(11)
1344          $ - 9, F
1345      Y,  $ + :MT[3]
1346      Y,  $ = MS
1347      Y,  MG[0] = $
1348          F + 1
1349          JUMP(- 9)
1350          64
```

```
1351            65
1352            88
1353            89
1354            93
1355            E = UCLEAR
1356            MC = E
1357            S = :RNX[UA1]
1358            SUBCD(:RSE77)
1359            S = :RNY[UA1]
1360            SUBCD(:RSE77)
1361            S = :RNANGLE[UA1]
1362            SUBC(:RSE85)
1363            S = :RNHEIGHT[UA1]
1364            SUBC(:RSE85)
1365            S = :RNITALICITY[UA1]
1366            SUBC(:RSE85)
1367            S = :RNFIRST[UA1]
1368            SUBC(:RSE94)
1369            S = :RN[UA1]
1370            SUBC(:RSE78)
1371            S = :RNNUMBER[UA1]
1372            SUBC(:RSE85)

1373            S = 41
1374            SE73
1375            UINVPLOTTEXT
1376            S = PLOTNUMBERPRICE
1377            GBILL + S, P
1378            SE115
1379            SE71
1380            UINVABSE:XRLOT

1381 CONVERTNUM(9)
1382            S = MC[-1]
1383            CGL2 = S
1384            SUBC(:MT[5])
1385            SUBC(:MT[-256])
1386            S = CGL2
1387            MC = S
1388            DO(MD[-1])
1389            CTERUG
1390            JUMP(-'505')
1391 UINVBEGINRLDOC(218)
1392            S = MC[-1]
1393            GL3 = S
1394            S = 5
1395            SE0
1396            S = 238
1397            SE1
1398            SUBCD(:UIMMORTAL)
1399            UDEAE
1400            G = UCLAIMRLO
1401            G + ULOANRLO
1402            G - UDOCRLO
1403            E - 0, P
1404      N,    SUBCD(:UMORTAL)
1405      N,    A = 235
1406      N,    SUBCD(:UDYNERR)
1407            G = UBLOCKSTR
1408            MG[6] = B
```

call number conversion

" CALL CE:XRLO

" SAVE INVARIANT RETURN ADDRESS

" ASK SUFFICIENT STACK SPACE FOR IDENTIFICATION

" NUMBER OF DOCUMENTS UNDER CONSTRUCTION < SPECIFIED NEED?

" DOCUMENT UNDER CONSTRUCTION := TRUE

```
1409        S = 1
1410        UDOCPLO + S                    " INCREASE NUMBER OF DOCUMENTS UNDER CONSTRUCTION
1411        UFIRSTENTRANCE = - B
1412        USTATE1 = B
1413        USTATE2 = B
1414        USTATE4 = B
1415        USTATE8 = B
1416        USTATE16 = B
1417        USTATE32 = - B
1418        USTATE64 = - B
1419        RREN = B
1420        F = 0
1421        RXMIN = F
1422        RYMIN = F
1423        RXLAST = G
1424        F = URLOWIDTH
1425        RXMAX = G
1426        RYMAX = G

1427        RYLAST = G
1428        F = 1
1429        RSCX = F
1430        RSCY = F
1431        B + 2
1432        DOS(RSE76)
1433        + 0
1434        + 50
1435        DOS(RSE76)
1436        + 0
1437        URLOWIDTH(- 50)
1438        DOS(RSE76)
1439        + 0
1440        + 270
1441        DOS(RSE76)
1442        - 0
1443        - 50
1444        DOS(RSE76)
1445        + 0
1446        + 0
1447        DOS(RSE89)
1448        + 0
1449        + 0
1450        DOS(RSE76)
1451        + 0
1452        + 1
1453        DOS(RSE76)
1454        + 0
1455        + 4
1456        DOS(RSE76)
1457        + 0
1458        + 0
1459        S = :URUNNUM
1460        SUBC(:RSE78)
1461        S = 49
1462        SE73
1463        UINVABSFIXPLOT

1464        B + 2                          " ABSFIXPLOT(0, 0, 270, - 50, 0, FALSE, 1, 2, 0, DOCNUMBER)
1465        DOS(RSE76)
1466        + 0
```

creates heading and identification of a plotter picture

" ABSFIXPLOT(50, 2744, 270, - 50, 0, TRUE, 1, 4, 0, RUNNUMBER)

```
1467        + 0
1468        DOS(RSE76)
1469        + 0
1470        + 0
1471        DOS(RSE76)
1472        + 0
1473        + 270
1474        DOS(RSE76)
1475        - 0
1476        - 50
1477        DOS(RSE76)
1478        + 0
1479        + 0
1480        DOS(RSE89)
1481        - 0
1482        - 0
1483        DOS(RSE76)
1484        + 0
1485        + 1
1486        DOS(RSE76)
1487        + 0
1488        + 2
1489        DOS(RSE76)
1490        + 0
1491        + 0
1492        G = VBLOCKSTR
1493        S = :MG[9]
1494        SUBC(:RSE78)
1495        S = 49
1496        SE73
1497        UINVABSFIXPLOT
1498        B + 2                    " ABSFIXPLOT(U, 0, 270, - 50, 0, FALSE, 1, 6, 0, DATE)
1499        DOS(RSE76)
1500        + 0
1501        + 0
1502        DOS(RSE76)
1503        + 0
1504        + 0
1505        DOS(RSE76)
1506        + 0
1507        + 270
1508        DOS(RSE76)
1509        - 0
1510        - 50
1511        DOS(RSE76)
1512        + 0
1513        + 0
1514        DOS(RSE89)
1515        - 0
1516        - 0
1517        DOS(RSE76)
1518        + 0
1519        + 1
1520        DOS(RSE76)
1521        + 0
1522        + 6
1523        DOS(RSE76)
1524        + 0
1525        + 0
```

```
1526        S = :UDATE
1527        SUBC(:RSE78)
1528        S = 49
1529        SE73
1530        UINVABSE!XPLOT

1531        B + 2                          " PLOTTEXT(150, 2744, 270, - 50, 0, TRUE, 1, NAME)
1532        DOS(RSE76)
1533        + 0
1534        + 150
1535        DOS(RSE76)
1536        + 0
1537        URLOWIDTH[- 50]
1538        DOS(RSE76)
1539        + 0
1540        + 270
1541        DOS(RSE76)
1542        - 0.
1543        - 50
1544        DOS(RSE76)
1545        + 0
1546        + 0
1547        DOS(RSE89)
1548        + 0
1549        + 0
1550        DOS(RSE76)
1551        + 0
1552        + 1
1553        G = URROGID
1554        SUBC(:RSE112)
1555        S = 41
1556        SE 73
1557        UINVRLOTTEXT

1558        B + 2                          " PLOT(250, 0, - 2)
1559        DOS(RSE76)                       pen positioning
1560        + 0
1561        + 250
1562        DOS(RSE76)
1563        + 0
1564        + 0
1565        DOS(RSE76)
1566        - 0
1567        - 2
1568        S = 21
1569        SE73
1570        UINVPLOT
1571        S = UBLOCKSTR
1572        F = - 0
1573        MS[19] = F
1574        MS[21] = F
1575        UFIRSTENTRANCE = B
1576        F = UCLEAR
1577        USTATE1 = G
1578        USTATE2 = G
1579        USTATE4 = G
1580        USTATE8 = G
1581        USTATE16 = G
1582        USTATE32 = G
1583        USTATE64 = G
```

```
1584          RXLAST = G
1585          RYLAST = G
1586          RXMAX = G
1587          RYMAX = G
1588          RREN = G
1589          RSCX = F
1590          RSCY = F
1591          RXMIN = F
1592          RYMIN = F
1593          RCY1R = F
1594          RCSCALEF = F
1595          RCX1 = F

1596          RCX2 = F
1597          RCY1 = F
1598          RCY2 = F
1599          RCCOUNT = G
1600          RCDERIV = G
1601          RCXISX = G
1602          RTX0 = F
1603          RTY0 = F
1604          RTSHIFT = G
1605          SE6
1606          SUBCD(:UMORTAL)
1607          S = GL3
1608          MC = S
1609          CTERUG

1610 GLINENUMBER(14)
1611          A = 1
1612          S = 4
1613          SE18
1614          S = MA[ - 6]
1615      U,  S - 9, Z
1616      N,  SUBCD(:RSE19)
1617          S = ULRSTREAMS
1618          S = MS
1619          G = MS[13]
1620          R + 1
1621          S = 12
1622          GBILL + S, R
1623          SE115
1624          SE71

1625 UJOB3(6)
1626      U,  S - 132, Z                    " RUNSTATISTICS ?
1627      Y,  JUMP(2)
1628          SE33
1629          UNINT
1630          SE33
1631          CRUNSTAT
1632 CRUNEMPTY(7)
1633          '033 036 027'                 " RUN
1634          '034 035 012'                 " STA
1635          '035 022 034'                 " TIS
1636          '035 022 014'                 " TIC
1637          '034 135 016'                 " S E
1638          '026 031 035'                 " MPT
1639          '042 130 776'                 " Y,
1640 CINVRUNSTAT(5)
```

```
1641          '033 036 027'
1642          '034 035 012'
1643          '035 022 034'
1644          '035 022 014'
1645          '034 135 776'                    " RUNSTATISTICS
1646 CRUNSEG(1)
1647          CRUNSEG
1648 CRUNSTAT(186)
1649          S = 3
1650          UKIND = S
1651          UDEAF
1652          SUBCD(:UCLAIMCONRED)
1653          UTREXR
1654          S = - USRECIALRM, R
1655      Y,  S = URUNACCEPT, Z
1656          G = MT[9]                        " NAME RUNSTATISTICS
1657          A = 1
1658      N,  JUMP(9)
1659          UWHO = G
1660          UTREXV
1661          F = - 0
1662          MC = F
1663          B + 3
1664          SE33
1665          UNAC
1666          CINVRUNSTAT
1667          UINVMBEGIN
1668          UPROGID = G
1669          UKILLVAR = A, E
1670      Y,  UCLAIMRUN = A
1671          S = UCURRUNNUM
1672          URUNNUM = S
1673          UCURRUNNUM + A
1674          UTREXV
1675          S = :M0[0]
1676          SUBCD(:UPROGNAME)
1677          G = MT[ - 11]
1678          SUBCD(:CPRINTSTRING)
1679          SUBCD(:UCONFREEWHITE)
1680          DO(MD[ - 1])

1681          S = 10
1682          SE0
1683          S = 50
1684          SE1
1685          SUBCD(:UIMMORTAL)
1686          F = :URUNSEM
1687          UP
1688          S = MT[8]
1689      U,  S - URUNAAN, Z
1690      N,  JUMP(9)
1691          SUBCD(:UCLAIMCONRED)
1692          G = MT[5]                        " RUNSTATISTICS EMPTY
1693          SUBCD(:CPRINTSTRING)
1694          SUBCD(:UCONFREEWHITE)
1695          DO(MD[ - 1])
1696          JUMP(128)
1697          UINVRUN1[ - 1]
1698          CRUNEMPTY
1699          CRUNKOP
```

this program dumps runstatistics of finished programs on request of the operator.

```
1700          S + 1
1701          MA[2] = S
1702          G = MT[ - 4]
1703          B + 2
1704          SE112
1705          S = 13
1706          SE73
1707          CPRINTTEXT                    " PRINT PAGEHEADING
1708          MA[3] = B                     " BEGIN := IRUE
1709          B + 2
1710          S = 9
1711          SE73
1712          CNLCR
1713          A = MA[2], Z
1714          SUBCD(:RSE39)                 " READ SERIAL NUMBER
1715          MA[4] = G                     " FILL SERIAL NUMBER
1716          CHEEN
1717          CRUNABS
1718          S = 20

1719          MA[5] = S
1720          G = MA[2]
1721          F + 1
1722          SUBCD(:UINSTV)
1723          R = :MC[ - 3]
1724          SUBCD(:UDNSS)
1725          S = A
1726          DO(MD[ - 1])
1727          MA[6] = S                     " SAVE NEXT STRING SYMBOL
1728     U,   S - 510, Z                    " END OF STRING ?
1729     Y,   JUMP(19)
1730          S - 9, P                      " - DIGIT ?
1731     Y,   MA[3] = - B                   " BEGIN := EALSE
1732          S = MA[3], P                  " BEGIN ?
1733     N,   JUMP(6)
1734          B + 2
1735          S = :MA[6]
1736          SUBC(:RSE78)
1737          S = 13
1738          SE73
1739          CRUSYM1
1740          B + 2
1741          S = :MA[6]
1742          SUBC(:RSE78)
1743          S = 13
1744          SE73
1745          CRRSYM
1746          S = 1
1747          MA[5] - S
1748          JUMP( - 26)
1749          B + 2
1750          S = :MA[5]
1751          SUBC(:RSE78)
1752          S = 13
1753          SE73
1754          CSPACE
1755          CHEEN
1756          CRUNSTAT[183]

1757          A = MA[2]
```

```
1758          A + 8, Z
1759          SUBCD(:RSE39)
1760          MA[4] = G
1761          F - 4
1762          F * 4
1763          G + MT[6]
1764          B + 2
1765          SE112
1766          S = 13
1767          SE73
1768          CPRINTTEXT
1769          JUMP(1)
1770          CAELOOP
1771          S = MA[4]
1772          S - 4, Z
1773      N,  S - 2, Z
1774      N,  S = 1
1775          MA[4] = S
1776          CWEEN
1777          CRUNABS[2]
1778          A = MA[2]
1779          A + 10, Z
1780          SUBCD(:RSE39)                    " TRANSLATION TIME
1781          MC = G
1782          A = MA[2]
1783          A + 11, Z
1784          SUBCD(:RSE39)                    " EXECUTION TIME
1785          G + MC[ - 1]
1786          MA[4] = G
1787          CWEEN
1788          CRUNABS
1789          CWEEN
1790          CRUNABS[2]                       " PUNCH TIME
1791          A = MA[2]
1792          A + 9, Z
1793          SUBCD(:RSE39)                    " NUMBER OF PRINTER FORMS
1794          MA[4] = G
1795          CWEEN
1796          CRUNABS
1797          CWEEN
1798          CRUNABS[2]

1799          A = MA[2]
1800          A + 12, Z
1801          SUBCD(:RSE39)
1802          MA[4] = G                        " NUMBER OF PLOTTERSTEPS
1803          CWEEN
1804          CRUNABS
1805          CWEEN
1806          CRUNABS[2]
1807          S = 2
1808          MA[4] = S                        "PUNCH 2
1809          CWEEN
1810          CRUNABS[2]
1811          S = UDATE
1812          MA[4] = S                        "PUNCH DATE
1813          CWEEN
1814          CRUNABS[2]
1815          B + 2
1816          S = 9
```

```
1817          SE73
1818          CRUNLCR1                          "RUNCM NLCR
1819          $ = 13
1820          PLUS$(MA[2])
1821          $ - URUNAAN, P                    " FINISHED ?
1822      N,  JUMP(-115)
1823          $ = MT[-127]                      "UINVRUN1[-1]
1824          URUNAAN = $
1825          F = :URUNSEM
1826          UV
1827          SUBCD(:UMORTAL)
1828          SE6
1829          $ = 0
1830          $E33
1831          UNORMALEXIT
1832          B + 1
1833          $E33
1834          CRUNABS[18]

1835 CRUNABS(30)
1836          $ = 0                             " FOR PRINT
1837          JUMP(1)
1838          $ = 1                             " FOR PUNCH
1839          MC = $
1840          B + 2
1841          DOS(RSE76)
1842          0
1843          8
1844          DOS(RSE76)
1845          0
1846          0
1847          $ = :MA[4]
1848          SUBC(:RSE78)
1849          $ = 21
1850      U,  $ = M[B - 15], Z                  " PRINT ?
1851      Y,  JUMP(10)
1852          SE73
1853          CABSFIXR1
1854          B + 2
1855          DOS(RSE76)
1856          0
1857          87
1858          $ = 13
1859          SE73
1860          CRUSYM1
1861          JUMP(2)
1862          SE73
1863          CABSFIXT
1864          B - 1
1865          CTERUG

1866 CAFLOOP(20)
1867          '030 031 016'                     " OPE
1868          '033 012 035'                     " RAT
1869          '016 036 033'                     " EUR
1870          '135 776 000'
1871          '016 041 016'                     " EXE
1872          '014 036 035'                     " CUT
1873          '022 016 135'                     " IE
1874          '135 776 000'
```

```
1875          '027 030 033'                    " NOR
1876          '026 012 012'                    " MAA
1877          '025 135 135'                    " L
1878          '135 776 000'
1879          '031 012 034'                    " PAS
1880          '034 135 001'                    " S 1
1881          '135 135 135'
1882          '135 776 000'
1883          '031 012 034'                    " PAS
1884          '034 135 002'                    "S 2
1885          '135 135 135'
1886          '135 776 135'
1887 CRUNKOP(24)
1888          '135 135 034'                    "   S
1889          '016 033 022'                    " ERI
1890          '016 027 033'                    " ENR
1891          '135 022 015'                    "  ID
1892          '016 027 035'                    " ENT
1893          '022 017 022'                    " IFI
1894          '014 012 035'                    " CAT
1895          '022 016 135'                    " IE
1896          '135 135 135'
1897          '135 135 135'
1898          '012 017 025'                    " AFL
1899          '030 030 031'                    "OOP
1900          '135 135 135'
1901          '135 135 135'
1902          '135 135 135'
1903          '035 022 023'                    " TIJ
1904          '015 135 135'                    " D
1905          '135 031 033'                    " PR
1906          '022 027 035'                    " INT
1907          '016 033 135'                    " ER
1908          ' 135 135 031'                   "  P
1909          '025 030 035'                    " LOT
1910          '035 016 033'                    " TER
1911          '167 776 000'                    " NLCR
```

```
 0 INAL(20)
 1          A=1
 2          $=4
 3          SE18
 4          $=MA[-6]
 5 U,       $-9,Z
 6 N,       SUBCD(:RSE19)
 7          JUMP(49)
 8          $'*'UARCKB[1],Z
 9 Y,       F=-0
10 Y,       JUMP(6)
11          F=1
12          $=2
13          UARCKB[1]=$
14          DO(UIETRC[4])
15 U,       $-UARCKB[1],Z
16 N,       JUMP(-4)
17          $=11
18          GBILL+$,R
19          SE115
20          SE71

21 UTELESTRING(31)
22          A=1
23          $=4
24          SE18
25          $=MA[-6]
26 U,       $-13,Z
27 N,       SUBCD(:RSE19)
28          F=:MA[-11]
29          SUBCD(:RSE29)
30          $=19
31          SE1
32          G=MA[-9]
33          SUBCD(:UINSTV)
34          F=:MC[-3]
35          SUBCD(:UDNSS)
36          MC=A
37 U,       A-510,Z
38          DO(MD[-1])
39 Y,       JUMP(8)
40          B+2
41          $=:MC[-3]
42          SE78
43          $=13
44          SE73
45          KARAKTERUIT
46          B-1
47          JUMP(-14)
48          B-4
49          $=11
50          GBILL+$,R
51          SE115
52          SE71

53 INAL2(7)
54          UDEAE
55          $=:MD[0]
56 U,       $-CONSOLECLAIMED,Z
```

procedures to read characters from and print characters on a conversational terminal

These procedures may be called from an algol program.

(I wrote a machinecode simulator in ALGOL using this device)

```
"GEEN PARAMETERS

"TEST CLAIM
"IRT=0?


"MAAK IET NUL


"KOSTEN









"IS AANTAL PARAMETERS 1

"IS PARAMETER VAN HET TYPE
"STRING?
"RESERVEER
"STAPELRUIMTE
"INVARIANT STARTING ADDRES



"EINDE STRING?


"MEEGEVEN INTEGER
"WAARDE AAN KARAKTERUIT
"EEN PARAMETER
```

```
57 N,       A=268
58 N,       $UBCD(:UDYNERR)
59          S=-UCON2                          "S:=-(2↑18-1)
60          JUMP(-56)                         "TERUG NAAR INAL

61 JMTRROC1(1)
62          JMTRROC1
63 JINSRTAPE(17)
64          A = 1
65          S = 4
66          SE18
67          S = 13
68      U,  S - MA[-6],  Z
69      N,  SUBCD(:RSE19)
70          F = :MA[-11]
71          SUBCD(:RSE20)
72          S = 16
73          SE1
74          DOS(MA[-11])
75          DO(ERV)
76      U,  S = F,  Z
77      N,  SUBCD(:RSE88)
78          SUBC(:CINSRTAPE)
79          S = 15
80          JUMP(21)                          " TO JWRITETAPE[21]
81 JWRITETAPE(24)
82          A = 1
83          S = 4
84          SE18
85          S = 17
86      U,  S - MA[-6],  Z
87      N,  SUBCD(:RSE19)
88          F = :MA[-15]
89          SUBCD(:RSE20)
90          F = :MA[-11]
91          SUBCD(:RSE20)
92          S = 16
93          SE1
94          DOS(MA[-15])
95          DO(ERV)
96      U,  S = F,  Z
97      N,  SUBCD(:RSE88)
98          MC = G                            " REEL
99          DOS(MA[-11])
100         DO(ERV)
101         SUBC(:CWRITETAPE)
102         S = JMTREWRRR
103         GBILL + S, P
104         SE115
105         SE71
106 CWRITETAPE1(9)
107         G = JCURUNIT
108         SUBC(:MT[44])
109         A = 1
110         MG[16] = A                        " POS := 1
111         MG[19] + A                        " LASTELEMENT + 1
112         A = 2
113         MG[4] = A                         " TAPESTATUS := 2
114         SUBCD(:UMORTAL)
115         JUMP(33)
```

magnetic tape procedures

reading, writing of real numbers and positioning of tape.

Blocking is taken care of by the system.

```
116 JREADTAPE(16)
117           A = 1
118           S = 4
119           SE18
120           S = 13
121       U,  S - MA[-6],  Z
122       N,  SUBCD(:RSE19)
123           F = :MA[-11]
124           SUBCD(:RSE20)
125           S = 16
126           SE1
127           DOS(MA[-11])
128           DO(ERV)
129       U,  S = F,  Z
130       N,  SUBCD(:RSE88)
131           SUBC(:CREADTAPE)
132           JUMP( - 30)                    " TO EXIT JWRITETAPE
133 CREADTAPE1(19)
134           G = JCURUNIT
135           S = :MG[3]
136           SUBCD(:UKILLSEGR)
137           A = MG[21], R                  " READFORW ?
138       N,  SUBCD(:CMTRADRES)
139       N,  SUBC(:MT[103])                 " SPOELF
140       N,  SUBC(:MT[75])                  " BLOCKCHECK
141           SUBC(:MT[49])
142           A = MG[8], R                   " INFOBLOCK ?
143       N,  MG[17] = - B                   " NEXTEL := FALSE
144           A = 1
145       Y,  MG[16] = A                     " POS := 1
146           MG[19] + A                     " LASTELEMENT + 1; MERGE FROM CREADBACK1
147           SUBCD(:UMORTAL)
148           F = MC[ - 3]
149           A = MC[1]
150           MC = A
151           DO(MD[ - 1])
152           CTERUG

153 JWRITEINFOBL(36)
154           SUBC(:MT[1])
155           CTERUG
156           S = MG[16]                     " POS
157       U,  S - 1,  Z                      " BUFFER EMPTY ?
158       Y,  GOTOR(MC[-1])                  " THEN READY
159           RUS(1)
160           A = JTELBLOCKWRITE             " BROKWOORD
161           LCS(9)
162           A + S                          " BLOCKLENGTH IN BROKWOORD
163           A + '60'                       " INFOBLOCKMARKER
164           S = :MG[3]                     " PRIVATE SV
165           SUBCD(:UHOLYRMSEG)
166           S = MS
167           S = MS[2]                      " CORE LOCATION SEGMENT
168           MS = A                         " FILL BROKWOORD
169           S + 511
170           MS = A                         " FILL SECOND BROKWOORD
171           S - 511                        " VAKSREC
172           A = 512                        " NUMBER OF WORDS IN BLOCK
173           CMTR
```

```
174          SUBC(:MT[80])                    " JWRITE
175          S = :MG[3]
176          SUBCD(:UKILLSEGR)
177          A = MG[13],  R                   " END OF TAPE ?
178      N,  SUBCD(:CMTVADRES)
179      N,  A = 1
180      N,  MG[10] + A                       " BLOCKCOUNT + 1
181      N,  GOTOR(MC[-1])
182          SUBC(:MT[198])                   " JWRITETAREM
183          A = MG[10]
184          MG[9] = -A                       " BLOCKAANT := -BLOCKCOUNT
185          SUBC(:MT[179])                   " JWRITETELBL
186          A = 6                            " MERGE FROM JWRITETELBL
187          MG[4] = A                        " TAPESTATUS := 6
188          A = 411
189          JUMP(386)                        " ALARM EN V(MTSEM)

190 JREADINFOBL(19)
191          SUBC(:MT[1])
192          CTERUG
193          S = :MG[3]
194          SUBCD(:UWOLYRMSEG)
195          S = MS
196          S = MS[2]
197          A = 512
198          CMTR
199          SUBC(:MT[10])                    " JREADBL
200          SUBC(:MT[17])                    " JBLOCKCHECK
201          A = MG[8],  R                    " INFOBLOCK ?
202      Y,  BUS(9)                           " BROKWOORD IN S
203      Y,  S '*' '7777'                     " BLOCKLENGTH UIT BROKWOORD
204      N,  S = 0
205          MG[15] = S                       " FILL BLOCKLENGTH
206          MG[21] = B                       " READERRV := TRUE
207          S = :MG[3]
208          SUBCD(:UBROEANATION)
209          GOTOR(MC[-1])

210 JREADBL(8)
211          SUBC(:MT[42])                    " JREAD
212          S = JLABELMT[4]
213          S '*' UCON2                      " VAKSREC
214          S = MS                           " BROKWOORD
215      U,  S '*' '60',  Z
216      Y,  MG[8] = -B                       " INFOBLOCK := FALSE
217      N,  MG[8] = B                        " INFOBLOCK := TRUE
218          GOTOR(MC[-1])

219 JBLOCKCHECK(21)
220          A = MG[12],  R                   " TAPEMARK ?
221      N,  JUMP(9)
222          SUBC(:MT[178])                   " JREADTELBL
223          A = MG[8],  R                    " INFOBLOCK ?
224      Y,  A = 416
225      Y,  JUMP(353)                        " ALARM
226          A = 1
227          MG[11] + A                       " GROUPCOUNT + 1
228          A = 0
229          MG[10] = A                       " BLOCKCOUNT := 0
230          JUMP(8)
```

```
231          A = -MG[14],  P                  " GEEN ENEN IN GAP ?
232      Y,  MG[8] = B                        " INEOBLOCK := TRUE
233      N,  SUBC(:MT[15])                    " JSPOELB
234      N,  SUBC(:MT[166])                   " JREADTELBL
235      N,  A = MG[8],  P                    " INEOBLOCK ?
236      N,  JUMP(-11)
237          A = 1
238          MG[10] + A                       " BLOCKCOUNT + 1
239          CMTV
240          GOTOR(MC[-1])

241 JSPOELB(2)
242          SUBC(:MT[5])
243          CTERUG
244 JSPOELB(2)
245          SUBC(:MT[5])
246          CTERUG
247 JREWIND(27)
248          SUBC(:MT[5])
249          CTERUG
250 " JSPOELB(2)
251          S = '400'                        " D8
252          JUMP(20)
253 " JSPOELB(2)
254          S = '20400'                      " D8 ^ D13
255          JUMP(18)
256 " JREWIND(2)
257          S = '40 000'                     " D14
258          JUMP(16)
259 " JREWUNL(2)
260          S = '60 000'                     " D13 ^ D14
261          JUMP(14)
262 " JREAD(3)
263          JLABELMT[4] = S                  " VAKSREC
264          S = '2400'                       " D8 ^ D10
265          JUMP(3)
266 " JWRITE(14)
267          JLABELMT[4] = S                  " VAKSREC
268          S = '1000'                       " D9
269          S + MG[18]                       " EVENT WISSCHRYF
270          RCA(9)                           " VAKLENGTE MOD(512)
271          JLABELMT[5] = A                  " IN TWEEDE WOORD VAKSREC
272          A '*' 7
273          A + '400'
274          RCA(9)
275          JLABELMT[4] + A                  " EERST WOORD VAKSREC
276          A = 1
277          MG[6] + A                        " REWRCOUNT + 1
278          S + MG[0]                        " UNITNR IN CODEWOORD
279          JLABELMT[2] = S                  " ORDRACHTWOORD
280          SUBC(:MT[87])                    " JSTARTMT( TO BE FOLLOWED BY JSLOTWOORDMT )

281 JSLOTWOORDMT(78)
282          S = 0
283          MG[18] = S                       " WISSCHRYF := 0
284          S = JLABELMT,  P                 " SLOTWOORD
285      N,  MC = S
286          MG[12] = -F                      " TAPEMARK := END OF TAPE := FALSE
287          A = JLABELMT[2]                  " ORDRACHTWOORD
288      Y,  JUMP(43)
```

```
289     U,  S + UCON1,  R
290         MC = A                          " SAVE ORDRACHTWOORD
291         S = 0
292         JLABELMT[2] = S                 " PREPARE FOR MOK
293         SUBC(:MT[75])                   " JSTARTMT
294         A = MC[-1]                      " OLD ORDRACHTWOORD
295         JLABELMT[2] = A                 " RESTORE IT
296     N,  JUMP(2)
297         SE33
298         JMTNBK
299         S = MC[-1]                      " SLOTWOORD
300     U,  S '*' 64,  Z                    " ¬ REWIND STATUS ?
301     N,  SUBC(:MT[147])                  " JREWINDMES
302     N,  JUMP(-22)                       " TO JWRITE[13]
303     U,  S '*' '200',  Z                 " ME ONDER CONTROLE ?
304     N,  S = 3
305     N,  MG[2] = S                       " UNITSTATUS := 3
306     N,  SUBC(:MT[139])                  " JOUTOECONMES
307     N,  S = 2
308     N,  MG[2] = S                       " UNITSTATUS := 2
309     N,  JUMP(-29)                       " TO JWRITE[13]
310     U,  S '*' '400',  Z                 " ¬ TAPEMARK ?
311     N,  MG[12] = B                      " TAPEMARK := TRUE
312     U,  S '*' '1000',  Z                " ¬ END OF TAPE ?
313     N,  MG[13] = B                      " END OF TAPE := TRUE
314     U,  S '*' '4000',  Z                " WRITING PERMITTED ?
315     N,  RCA(10),  R                     " D9 INTO SIGN POSITION
316     N,  A = 410
317     N,  MG[5] = A                       " STORECODE > 0
318     N,  JUMP(270)                       " ALARM
319     U,  S '*' '20 003',  Z              " ¬ PARITY ERROR ?
320     Y,  JUMP(17)
321         A = MG[5],  Z                   " STORECODE = 0 (FIRST TIME PARI ERROR) ?
322     N,  JUMP(5)
323         MC = A                          " ERRORCOUNT := 0
324         A = JLABELMT[2]                 " ORDRACHTWOORD
325     U,  A '*' '2400',  Z                " SCHRYVEN ?
326     Y,  A '+' '400'                     " DAN WISSCHRYF
327         MG[5] = A                       " ORDRACHTWOORD IN STORECODE
328         A = JLABELMT[2]
329         A '*' '20 000', Z               " ¬ REVERSE ?
330     N,  JUMP( - 74)                     " SPOELF
331     Y,  JUMP( - 73)                     " SPOELB
332         RUA(8)                          " OK
333         A '*' '136',  Z                 " SPOELORDR ?
334     Y,  A =256
335     Y,  MG[18] = A                      " WISSCHRYF:= 256
336     Y,  A = MG[5],  R                   " SPOELORDR NA PARITY ERROR ?
337     Y,  JUMP(6)
338         A = MG[5],  Z
339     N,  MG[5] - A
340     N,  B - 1
341         RCS(6)
342         MG[14] = -S                     " EWEN IN GAP
343         GOTOR(MC[-1])                   " EXIT SLOTWOORDMT
344         S = 1
345         MG[7] + S                       " TOTAL ERRORCOUNT + 1
346         PLUSS(M[B - 1])                 " ERRORCOUNT + 1
347     U,  S - 9,  R                       " AL 10 KEER ?
348     N,  S = A
```

```
349      N,  JUMP(- 70)                          " TO JWRITE[12]
350          BUA(9)
351      U,  A - 1,  Z                           " SCHRYVEN ?
352      Y,  A = 414
353      Y,  JUMP(235)                           " ALARM
354      U,  S - 16, R
355      Y,  A = 415
356      Y,  JUMP(232)                           " ALARM
357          S = 256
358          PLUSS(MG[5])
359          JUMP(- 80)                          " TO JWRITE[12]

360 JEINTAPE(9)
361          SUBC(:MT[ - 192])                   " WRITEINFOBL
362          A = MG[10],  R
363      N,  MG[9] = A
364      Y,  MG[9] = -A                          " BLOCKAANT := - ABS(BLOCKCOUNT)
365          CMTR
366          SUBC(:MT[28])                       " JWRITETAPEM
367          SUBC(:MT[11])                       " JWRITETELBL
368          CMTV
369          CTERUG

370 JSTARTMT(7)
371          S = UCON1                           " 2 ↑ 18
372          UARMT[2] = S
373          '760 070 000'(UNRMT)                " AE := TRUE
374          E = :UIETMT
375          URM                                 " R(IET)
376          G = JCURUNIT
377          GOTOR(MC[-1])

378 JWRITETELBL(16)
379          SUBC(:MT[1])
380          CTERUG
381          S = :JTELBLOCKWRITE                 " VAKSPEC
382          A = MG[9]
383          MS[1] = A                           " BLOCKAANT IN TELBLOCK
384          A = 3
385          SUBC(:MT[-115])                     " JWRITE
386          A = MG[13],  R                      " END OF TAPE ?
387      N,  A = 0
388      N,  MG[10] = A                          " BLOCKCOUNT := 0
389      N,  GOTOR(MC[ - 1])
390          A = MG[9], R
391      N,  JUMP( - 190)                        " MERGE WITH JWRITEINFOBL
392          MG[9] = - A
393          SUBC(;MT[ - 132])                   " JSPOELB
394          JUMP( - 193)                        " MERGE

395 JWRITETAPEM(12)
396          SUBC(:MT[1])
397          CTERUG
398          S = :JTAPEMWRITE
399          JLABELMT[4] = S
400          A = 3
401          S = '1040'                          " CODE SCHRYF IBM-BCD
402          SUBC(:MT[-129])                     " TO JWRITE[2]
403          A = MG[13],  R                      " END OF TAPE ?
404      Y,  JUMP(-205)                          " TO JWRITEINFOBL[29]
```

```
405           A = 1
406           MG[11] + A                        " GROUPCOUNT + 1
407           GOTOR(MC[-1])

408 JSPOELFORW(5)
409           SUBC(:MT[1])
410           CTERUG
411           CMTR
412           SUBC(:MT[-151])                   " JSPOELF
413           JUMP(-179)                        " BLOCKCHECK

414 JREADTELBL(8)
415           SUBC(:MT[1])
416           CTERUG
417           S = :JTELBLOCKREAD                " VAKSPEC
418           A = 3
419           SUBC(:MT[-192])                   " JREADBL
420           A = JTELBLOCKREAD[1]              " INHOUD TELBLOCK
421           MG[9] = A                         " NAAR BLOCKAANT
422           GOTOR(MC[-1])
423 JSPOELBACKW(29)
424           SUBC(:MT[1])
425           CTERUG
426           CMTR
427           SUBC(:MT[-162])                   " JSPOELB
428           A = 1
429           MG[10] - A                        " BLOCKCOUNT - 1
430      U,   A + MG[10],  Z                    " BLOCKCOUNT = -1?
431      N,   MG[8] = B                         " INFOBLOCK := TRUE
432      N,   JUMP(18)
433           SUBC(:MT[-16])                    " JREADTELBL
434           A = MG[8],  R                     " INFOBLOCK ?
435      Y,   A = 416
436      Y,   JUMP(159)                         " ALARM
437           SUBC(:MT[-172])                   " JSPOELB
438           SUBC(:MT[-173])                   " JSPOELB
439           SUBC(:MT[-176])                   " JSPOELF
440           A = MG[12],  R                    " TAPEMARK ?
441      Y,   SUBC(:MT[-176])                   " JSPOELB
442           A = 1
443           MG[11] - A                        " GROUPCOUNT - 1
444           S = MG[9],  R
445      N,   S = -S
446           MG[9] = S
447           S + 0,  Z
448      N,   S - 1
449           MG[10] = S                        " BLOCKCOUNT := BLOCKAANT - 1
450      N,   SUBC(:MT[-185])                   " JSPOELB
451           CMTV
452           GOTOR(MC[-1])

453 JOUTOFCONMES(3)
454           A = MT[1]                         " OUT OF CONTROL
455           JUMP(2)
456           JMTMESOOC
457 JREWINDMES(23)
458           A = MT[21]                        " WAITING FOR REWIND
459           S = JLABELMT[2]
460           MC = S
461           B = JLABELMT[4]
```

```
462         MC = E
463         CMTV
464         S = MG[0]                       " UNIT NUMBER
465         MC = S
466         MC = A
467         SUBCD(:UCLAIMCONRED)
468         G = MC[- 1]
469         SUBCD(:CRRINTSTRING)
470         A = MC[ - 1]
471         SUBCD(:CRRINTDEC)
472         SUBCD(:USTORREN[1])
473         CMTR
474         E = MC[ - 2]
475         JLABELMT[4] = E                 " RESTORE VAKSPECIFICATIE
476         S = MC[ - 1]
477         JLABELMT[2] = S                 " RESTORE OPDRACHTWOORD
478         G = JCURUNIT
479         GOTOR(MC[- 1])
480         JMTMESREW

481 JREADB(5)
482         SUBC(:MT[1])
483         CTERUG
484         JLABELMT[4] = S
485         S = '22 400'
486         JUMP( - 206)                    " TO JWRITE[3]
487 JNEWREEL(25)
488         SUBC(:MT[ - 215])               " JREWUNL
489         SUBC(:MT[2])
490         SUBC(:MT[ - 219])               " REWIND
491         CTERUG
492         CMTV
493         A = 5
494         MG[2] = A
495         SUBCD(:UCLAIMCONRED)
496         G = MT[14]
497         SUBCD(:CRRINTSTRING)
498         A = - M1[ - 15]                 " REELNR
499         SUBCD(:CRRINTDEC)
500         S = JCURUNIT
501         MS[1] = A                       " FILL IN REELNUMBER
502         G = MT[9]
503         SUBCD(:CRRINTSTRING)
504         A = MS[0]                       " UNIT NUMBER
505         SUBCD(:CRRINTDEC)
506         SUBCD(:USTORREN[1])
507         CMTR
508         A = 2
509         MG[2] = A                       " RESET UNIT STATUS
510         GOTOR(MC[ - 1])
511         JLOADREEL
512         JONUNIT

513 JMOVETAPE(39)
514         S = M1[-15],  Z                 " GROUP
515         MC = S
516     N,  S + 0,  R
517     N,  A = 408
518     N,  JUMR(82)                        "ALARM
519         S - MG[11],  R                  " GROUP > GROUPCOUNT ?
```

```
520     N,  S + 0,  Z                   " GROUP = GROUPCOUNT ?
521     Y,  S = -MG[9],  R              " VOORBIJ SLUITBLOCK ?
522     Y,  JUMP(10)
523         S = M[B - 1]
524         S - MG[11],  Z               " GROUP = GROUPCOUNT ?
525     Y,  JUMP(17)
526         S + 0,  R                   " GROUP > GROUPCOUNT ?
527     N,  JUMP(7)
528         SUBC(:MT[-112])             " JSPOELFORW
529         A = MG[8],  R               " INFOBLOCK ?
530     Y,  JUMP(-3)
531         A = MG[9],  R               " = SLUITBLOCK ?
532     Y,  JUMP(-10)
533         A = 1
534         JUMP(16)
535         SUBC(:MT[- 106])                    " JSPOELBACKW
536         A = MG[8],  R               " INFOBLOCK ?
537     Y,  JUMP(-3)
538         JUMP(-16)
539         S = M[B - 1]
540         S - MG[11],  Z               " GROUP = GROUPCOUNT ?
541     N,  A = 2
542     N,  JUMP(8)
543         S = M[B - 3]
544         S - MG[10],  Z               " BLOCK = BLOCKCOUNT ?
545     Y,  JUMP(4)
546         S + 0,  R                   " BLOCK > BLOCKCOUNT ?
547     Y,  SUBC(:MT[- 131])            " JSPOELFORW
548     N,  SUBC(:MT[-119])             " JSPOELBACKW
549         JUMP(-11)
550         A = 0
551         B - 1
552         CTERUG
553 JALJUMP(3)
554         CMTV
555         SE33
556         JALARM

557 JMTPROC2(1)
558         JMTPROC2
559 JORENTAPE(133)
560         A = 1
561         S = 4
562         SE18
563         S = 17
564     U,  S - MA[-6],  Z
565     N,  SUBCD(:PSE19)
566         E = :MA[-11]
567         SUBCD(:PSE29)               " CHECK FORMAL STRING
568         E = :MA[-15]
569         SUBCD(:PSE20)               " CHECK FORMAL ARITHM
570         S = 16
571         SE1
572         DOS(MA[-15])
573         DO(ERV)
574         SUBCD(:PSE88)               " ROUND REEL
575         SUBCD(:UNMMORTAL)
576         MA[-15] = -G  ,R            " REEL < 0 ?
577     N,  A = -MA[-15]  ,Z            " V REEL = 0 ?
578     Y,  A = 419
```

```
579     Y,  JUMP(475)                   " TO ALARM
580         $ = JMTCLAIMED
581     U,  $ - CMTUSED, P              " MORE UNITS ALLOWED ?
582     N,  A = 402
583     N,  JUMP(471)                   " TO ALARM
584         CMTP
585         B + 2
586         M[B - 2] = B                " INDICATION := # 0
587         $ =:JUNITLIST
588         G = M$, P
589     N,  JUMP(11)
590         A = MG[2], Z                " UNIT STATUS
591     Y,  M[B - 2] = F                " INDICATION := 0; M[B - 1] := SELECTED UNIT
592     Y,  JUMP(21)
593     U,  A - 1, Z
594     N,  JUMP(90)                    " TO TEST FOR REEL IN USE BY THIS RM
595     U,  A = M[B - 2], Z             " UNIT STATUS = 0 FOUND ALREADY ?
596     N,  M[B - 2] = F                " M[B - 1] := SELECTED UNIT
597     N,  M[B - 2] = A                " INDICATION := 1
598         A = MG[1]                   " REELNUMBER
599     U,  A + M1[ - 15], Z            " UNITSTATUS = 1 ^ REEL ON THIS UNIT ?
600     N,  JUMP(13)                    " TRY NEXT UNIT
601     N,  F = M[B - 2]
602         A = 2
603         MG[2] = A                   " UNITSTATUS := 2
604         A = :M0
605         MG[22] = A                  " INDICATION THIS RM
606         A = 1
607         MG[4] = A                   " TAPE STATUS = 4
608         CMTUSED + A
609         JCURUNIT = G
610         A = - M1[ - 15]             " A := REELNUMBER
611     Y,  JUMP(4)
612     U,  A = M[B - 2], Z             " SET CONDITION TO SELECTED UNIT HAD UNITSTATUS = 0
613         JUMP(432)                   " TO EXTENSION IN JORENTARE1
614         S + 1
615         JUMP( - 28)                 " TRY NEXT UNIT
616         CHEEN
617         JREADTELBL
618         CMTV
619     U,  A = MG[8],  P               " INFOBLOCK ?
620     Y,  A = 403
621     Y,  JUMP(433)
622     U,  A + M1[-15]  ,Z             " REELNR = REEL ?
623     N,  JUMP(420)
624         MG[1] = A                   " REELNR INTO UNIT ADMINISTR
625         CHEEN
626         JREADINFOBL
627     U,  S = -MG[8],  P              " ¬ INFOBLOCK ?
628     Y,  JUMP(-9)
629         CHEEN
630         JSPOELFORW
631     U,  S = MG[8],  P               " INFOBLOCK ?
632     Y,  JUMP(-13)                   " ALARM
633         A = MG[9],  P               " ¬ SLUITBLOCK ?
634         G = M1[-9]
635         SUBCD(:UINSTV)
636         G = JCURUNIT
637         S = :MG[3]
638         LCS(9)
```

```
639          $ + 1
640          MC = $
641      Y,  JUMP(18)
642          CHEEN
643          JSPOELBACKW
644          F = :MC[-4]                    " WRITE LABEL
645          SUBCD(:UDNSS)
646          G = A
647          SUBCD(:RSE50)
648          $ = 1
649          RLUSS(MC)
650          $ '*' 511,  Z                  " NAME TOO LONG ?
651      Y,  A = 404
652      Y,  JUMP(402)                      " ALARM
653          F - 510,  Z                    " END OF STRING ?
654      N,  JUMP(-11)
655          G = JCURUNIT
656          MG[16] = $                     " SET POS
657          CHEEN
658          JEINTAPE
659          JUMP(12)
660          A = M[B-1],  Z
661          SUBCD(:RSE39)
662          MC = G
663          F = :MC[-5]
664          SUBCD(:UDNSS)
665      U,  A - MC[-1],  Z
666      N,  A = 405
667      N,  JUMP(387)                      " ALARM
668          A - 510,  Z                    " END OF STRING ?
669      N,  A = 1
670          RLUSA(M[B - 1])
671      N,  JUMP(-11)
672          B - 4
673          A = JCURUNIT
674          F = 0
675          MA[6] = -F                     " REWRCOUNT := TOTALERRORCOUNT := 0
676          MA[19] = F                     " LASTGROUP := LASTELEMENT := 0
677          MA[17] = -G                    " NEXTEL := FALSE
678          MA[10] = F                     " BLOCKCOUNT := GROUPCOUNT := 0
679          S = JORENTAPERR
680          SUBCD(:UMORTAL)
681          GBILL + S,  F
682          SE115
683          DO(MD[-1])
684          SE71
685          A = MG[22]                     " UNIT STATUS > 1
686      U,  A - :M0, Z                     " ^ THIS RM
687      Y,  A = MG[1]
688      Y,  A + M1[ - 15], Z               " ^ THIS REEL ON UNIT ?
689      N,  JUMP( - 76)                    " TO $ + 1
690          CMTV
691          A = 401
692          JUMP(362)                      " ERROR : REPEATED CALL OF ORENTAPE FOR SAME REEL

693 JCLOSETAPE(100)
694          A = 1
695          $ = 4
696          SE18
697          $ = 17
```

```
698     U,  $ - MA[-6],  Z
699     N,  SUBCD(:RSE19)
700         E = :MA[-15]
701         SUBCD(:RSE20)
702         E = :MA[-11]
703         SUBCD(:RSE21)                   " CHECK FORMAL BOOLEAN
704         $ = 16
705         SE1
706         DOS(MA[-15])
707         DO(ERV)
708     U,  $ = E,  Z
709     N,  SUBCD(:RSE88)
710         MA[-15] = G
711         DOS(MA[-11])
712         DO(ERV)
713         MA[-11] = G
714         SUBCD(:UIMMORTAL)
715         A = MA[-15]
716         SUBC(:CREELCHECK[1])
717     N,  JUMP(63)
718         A = MG[6],  R                   " REWRCOUNT
719         G = MG[7]                       " TOTAL ERRORCOUNT
720         G / A                           " FRACTION PARITY ERRORS
721     Y,  E - MT[62],  R                  " > 2 ↑ -7 ?
722     Y,  SUBCD(:UCLAIMCONRED)
723         G = MT[59]
724     Y,  SUBCD(:CPRINTSTRING)
725         G = JCURUNIT
726     Y,  A = MG[1]                       " REELNR
727     Y,  SUBCD(:CPRINTDEC)
728         A = MG[7]
729         MG[7] - A                       " TOTAL ERRORCOUNT := 0
730     Y,  SUBCD(:CPRINTDEC)
731         A = MG[6]
732         MG[6] - A                       " REWRCOUNT := 0
733     Y,  SUBCD(:CPRINTDEC)
734     Y,  SUBCD(:UCONEREEWHITE)
735         G = JCURUNIT
736         A = MG[2]
737     U,  A - 2,  Z                       " UNITSTATUS = 2 ?
738     N,  JUMP(47)
739         A = MG[4]
740         A - 2,  Z
741     N,  A - 2,  Z
742     N,  A - 2,  Z                       " TAPESTATUS = 2,4 OF 6 ?
743     Y,  A = MG[5],  Z
744     N,  JUMP(2)
745         CHEEN
746         JEINTAPE
747         A = MG[5],  Z                   " MERGE FROM ALARM
748         MC = A
749     N,  MG[5] - A
750         CMTR
751         CHEEN
752         JREWIND
753         $ = M[B - 1],  Z
754     Y,  $ = M1[ -11],  R                " EREE ?
755     N,  JUMP(11)
756         CHEEN
757         JSROELE
```

```
758          CHEEN
759          JSROELF
760          S = -1
761          MG[9] = S
762          CHEEN
763          JEINTARE[5]
764          CMTR
765          CHEEN
766          JREWIND
767          CMTV
768          S = MC[-1],  Z                    " STORECODE = 0 ?
769      Y,  S = 1
770      N,  S = -0
771      Y,  A = MG[1]
772      N,  A = -0
773          MG[2] = S
774          MG[1] = A
775          A = 1
776          CMTUSED - A
777          A = - 0
778          MG[5] = A
779          S = :MG[3]
780          SUBCD(:UKILLSEGR)
781          S = JCLOSETARR
782          JUMP( - 102)                      " TO GENERAL PROCEDURE EXIT
783          JMTERRORMES
784          '777 400 000'                     " FLOATING VALUE 2 ↑ - 7
785          1
786          A - 3, Z                          " UNIT STATUS = 3 ?
787      Y,  S = 1
788      Y,  JMTEREE - S
789      Y,  S = 4
790      Y,  JUMP( - 18)
791          A + 2, Z                          " UNIT STATUS = 1 ?
792      N,  JUMP( - 23)
793          JUMP( - 13)

794 JTARESERR(2)
795          S = 0
796          JUMP(1)
797 JTARESERE(86)
798          S = 1
799          CGL1 = S
800          A = 1
801          S = 4
802          SE18
803          S = 9
804      U,  S - MA[-6],  Z
805      N,  SUBCD(:RSE19)
806          S = 20
807          SE1
808      U,  S = JMTCLAIMED,  R
809      N,  JUMP(60)                          " TO PROCEDURE EXIT
810      U,  S = CGL1, Z
811      N,  JMTEINDR = B
812          A = 3
813          SUBC(:MT[58])                     " STATUSCHECK(3)
814      N,  JUMP(9)
815          DO(MD[ - 1])
816          B + 2
```

```
817             G = MT[5]
818             SUBC(:RSE112)
819             S = 13
820             SE 73
821             CPRINTTEXT
822             JUMP(9)
823             JMTVERSLAGMES
824             A = 2
825             SUBC(:MT[46])
826         N,  JUMP(13)
827             S = VKILLVAR
828             S - 5, P                        " ¬ DYNAMISCHE FOUT ?
829         N,  S = - JMTEINDR, P                       " ∨ ¬ EINDRITEN ?
830         Y,  JUMP(12)
831             DO(MD[ - 1])
832             B + 2
833             G = MT[212]
834             SUBC(:RSE112)
835             S = 13
836             SE73
837             CPRINTTEXT
838             MC = B                          " INDICATIE := IBUE
839             JUMP(4)
840             A = 5
841             SUBC(:MT[30])
842         N,  JUMP(23)                        " TO PROCEDURE EXIT
843             MC = - B
844             A = 5
845             SUBC(:MT[26])
846         N,  A = 3
847         N,  SUBC(:MT[24])
848         N,  A = 2
849         N,  SUBC(:MT[22])
850         N,  JUMP(15)
851             S = M[B - 1], P             " INDICATIE ?
852         N,  JUMP(2)
853             CHEEN
854             JTAPEVERSLAG
855             S = :MG[1]
856             DO(MD[ - 1])
857             B + 2
858             SUBC(:RSE78)
859             DOS(RSE89)
860             - 0
861             - 0
862             S = 17
863             SE73
864             JCLOSETAPE
865             JUMP( - 22)
866             S = JMTCLAIMED
867             JMTFREE + S
868             JMTCLAIMED - S
869             JMTEINDR = - S
870             S = 15
871             JUMP( - 188)
872             S = :JUNITLIST              " LOCAL SUBROUTINE STATUSCHECK
873             G = MS, P
874         N,  GOTO(MC[ - 1])
875         U,  A - MG[2], Z
876         N,  S + 1
```

```
877      N,  JUMP( - 5)
878          A = :MC[0]
879      U,  A - MG[22], Z
880      N,  A = MG[2]
881      N,  JUMP( - 6)
882          JCURUNIT = G
883          GOTO(MC[ - 1])

884  JWRITEPOSITION(127)
885          S = 1
886          CGL1 = S
887          A = 1
888          S = 4
889          SE18
890          S = 21
891      U,  S - MA[ - 6], Z
892      N,  SUBCD(:RSE19)
893          F = :MA[-19]
894          SUBCD(:RSE20)
895          F = :MA[-15]
896          SUBCD(:RSE20)
897          F = :MA[-11]
898          SUBCD(:RSE20)
899          S = 16
900          SE1
901          S = CGL1
902          MC = S
903          DOS(MA[-19])
904          DO(ERV)
905      U,  S = F,  Z
906      N,  SUBCD(:RSE88)
907          MA[-19] = G                    " REEL
908          DOS(MA[-15])
909          DO(ERV)
910      U,  S = F,  Z
911      N,  SUBCD(:RSE88)
912          MA[-15] = G                    " GROUP
913          DOS(MA[-11])
914          DO(ERV)
915      U,  S = F,  Z
916      N,  SUBCD(:RSE88)
917          MA[-11] = G                    " ELEMENT
918          SUBCD(:UIMMORTAL)
919          A = MA[-19]
920          SUBC(:CREELCHECK[1])
921      N,  A = 406
922      N,  JUMP(136)                      " ALARM
923          A = MG[2]
924          A - 2,  Z                      " UNITSTATUS = 2 ?
925      N,  JUMP(-5)                       " ALARM
926          A = MG[4]
927          A - 2,  Z
928      N,  A - 2,  Z
929      N,  A - 2,  Z                      " TAPESTATUS = 2,4 OF 6 ?
930      N,  JUMP(2)
931          CHEEN
932          JEINTAPE
933          S = MC[-1],  Z                 " READPOSITION ?
934      Y,  JUMP(77)                       " TO JREADPOS1
935          S = M1[-11],  Z                " ELEMENT
```

```
936     N,  S - 1,  Z
937     Y,  S = 0
938     Y,  JUMP(5)
939         S + 0,  P
940     N,  A = 407
941     N,  JUMP(117)
942         A = 0
943         DIVAS(255)
944         MC = S                          " BLOCK
945         CHEEN
946         JMOVETAPE
947     U,  A - 2,  Z
948     Y,  JUMP(51)
949         A + 0,  Z
950     N,  JUMP(29)
951         CHEEN
952         JREADINFOBL
953         S = M[0 - 1]                    "BLOCK
954         MULS(255)
955         A = M1[-11]                     " ELEMENT
956         S + MG[15]                      " ADD BLOCKLENGTH
957     U,  A - S,  P                       " ELEMENT > BLOCK * 255 + BLOCKLENGTH ?
958     N,  JUMP(6)
959         A = MG[8],  P                   " INFOBLOCK ?
960     N,  JUMP(2)
961         CHEEN
962         JSRQELFORW
963         A = 2
964         JUMP(37)
965         S = M[0 - 1]                    " BLOCK
966         S + 1
967         MULS(255)
968         S - M1[-11],  Z                 " ELEMENT = ( BLOCK + 1 ) * 255 ?
969     Y,  JUMP(2)
970         CHEEN
971         JSRQELBACKW
972         A = M1[-11]                     " ELEMENT
973         S = MG[10]                      " BLOCKCOUNT
974         MULAS(-255)
975         LCS(1)
976         S + 1
977         MG[16] = S                      " POS := (ELEMENT - BLOCK * 255) * 2 + 1
978         A = 0
979         JUMP(22)
980         CMTP
981         CHEEN
982         JSRQELB
983         A = MG[9]
984         MG[9] = -A                      " BLOCKAANT := -BLOCKAANT
985         CHEEN
986         JWRITETELBL
987         CMTV
988         A = 1
989         MG[16] = A                      " POS := 1
990         S = 0
991         MG[10] = S
992         A = M1[-15]                     " GROUP
993         A - MG[11],  P                  " GROUP > GROUPCOUNT ?
994     N,  JUMP(4)
995         A = -0
```

```
 996          CHEEN
 997          JEINTAPE[3]
 998          JUMP(-7)
 999          A = 1
1000     U,   A = M1[-11],  Z                  " ELEMENT = 0 ?
1001     N,   A + 2
1002          S = :MA[3]
1003          MG[17] = -B                      " NEXTEL := FALSE
1004          MG[4] = S
1005          S = M1[-11]
1006          MG[19] = S                       " LASTELEMENT := ELEMENT
1007          S = M1[-15]
1008          MG[20] = S                       " LASTGROUP := GROUP
1009          G = A
1010          S = JPOSPRICE
1011          JUMP( - 328)                     " TO GENERAL EXIT
1012 JREADPOS1(45)
1013          S = M1[-11],  Z                  " ELEMENT
1014     Y,   JUMP(5)
1015          S + 0,  P
1016     N,   A = 407
1017     N,   JUMP(43)                         " ALARM
1018          A = 0
1019          DIVAS(255)
1020          MC = S                           " BLOCK
1021          CHEEN
1022          JMOVETAPE
1023          A + 0,  Z
1024     N,   S = 7
1025     N,   MG[17] = -B                      " NEXTEL := FALSE
1026     N,   JUMP( - 22)                      " MERGE WITH JWRITEPOSITION[119]
1027          CHEEN
1028          JREADINFOBL
1029          A = MG[8],  P                    " INFOBLOCK ?
1030     N,   A = 2
1031     N,   JUMP(-8)
1032          A = M1[-11]                      " ELEMENT
1033          S = M[B - 1]                     " BLOCK
1034          MULAS(-255)
1035          A = MG[15]
1036     U,   A - S,  P                        " BLOCKLENGTH > ELEMENT - BLOCK * 255 ?
1037     Y,   JUMP(3)
1038          CHEEN
1039          JSPOELFORW
1040          JUMP(-11)
1041          LES(1)
1042          S + 1
1043          MG[16] = S                       " SET POS
1044          A = 0
1045          S = 8
1046          MG[17] = B                       " NEXTEL := TRUE
1047          JUMP( - 43)                      " MERGE WITH JWRITEPOSITION[119]
1048          JMTVERSLAGKOP                    " FOR THE BENEFIT OF JTAPESERE[35]
1049          CMTP                             " EXTENSION FROM REEL # REELNR IN OPENTAPE
1050          JUMP(- 435)                      " TO CHEEN; JNEVREEL
1051     Y,   JUMP(3)                          " JOPENTAPE1
1052          CHEEN
1053          JNEVREEL
1054          JUMP( - 434)                     " RETURN TO READTELBL
1055          CHEEN
```

```
1056          JNEVREEL(1)
1057          JUMP( - 437)                    " RETURN

1058 JREADPOSITION(2)
1059          $ = 0
1060          JUMP( - 173)                    " TO JWRITEPOSITION(1)

1061 JALARM(16)
1062          $ = A
1063          DO(MD[-1])
1064          B = MA
1065      U,  $ = JMTENDR, P                  " EINDRITEN ?  ?
1066      N,  A = :MS[-152]
1067      N,  SUBCD(:UMORTAL)
1068      N,  SUBCD(:UDYNERR)
1069          B + 2
1070          G = MT[6]
1071          SUBC(:PSE112)
1072          $ = 13
1073          SE73
1074          CPRINTTEXT
1075          G = JCURUNIT
1076          JUMP( - 324)                    " MERGE WITH JCLOSETAPE[53]
1077          JEINDMES

1078 JMTPROC3(1)
1079          JMTPROC3
1080 JTAPEVERSLAG(37)
1081          DO(MD[-1])
1082          $ = :MG[1]
1083          B + 2
1084          SUBC(:PSE78)
1085          $ = 13
1086          SE73
1087          CPRNTX
1088          G = JCURUNIT
1089          $ = :MG[20]
1090          B + 2
1091          SUBC(:PSE78)
1092          $ = 13
1093          SE73
1094          CPRNTX
1095          G = JCURUNIT
1096          $ = :MG[19]
1097          B + 2
1098          SUBC(:PSE78)
1099          $ = 13
1100          SE73
1101          CPRNTX
1102          G = JCURUNIT
1103          $ = MG[4]
1104          $ - 2,  Z
1105      N,  $ - 2,  Z
1106      N,  $ - 2,  Z
1107      N,  E = MT[7]
1108      Y,  E = MT[7]
1109          B + 2
1110          SUBC(:PSE112)
1111          $ = 13
1112          SE73
```

```
1113          CPRINTTEXT
1114          G = JCURUNIT
1115          CTERUG
1116          JMTMODELEES
1117          JMTMODESCHRIJF

1118 JMTNBK(26)
1119          CMTV
1120          SUBCD(:UCLAIMCONRED)
1121          $ = :M0[0]
1122          SUBCD(:UPROGNAME)
1123          G = MT[13]
1124          SUBCD(:CPRINTSTRING)
1125          SUBCD(:UCONFREEWHITE)
1126          $ = :JUNITLIST
1127          G = MS,  R
1128      N,  JUMP(9)
1129          A = MG[2]
1130      U,  A - 2, Z
1131      Y,  A = :M0[0]
1132      Y,  A - MG[22], Z                    " ^ USED BY THIS PM
1133      Y,  A = 3
1134      Y,  MG[2] = A
1135          $ + 1
1136          JUMP( - 10)
1137          JMTNBKMES
1138      U,  $ = JMTEINDR, R
1139      Y,  JUMP(3)
1140          A = 418
1141          $E33
1142          JALARM
1143          $E33
1144          JCLOSETAPE[41]
1145 JINVM2(5)
1146          '027 030 035'                    " N O T
1147          '135 012 025'                    "   A L
1148          '025 030 040'                    " L O W
1149          '016 015 130'                    " E D .
1150          '776 000 000'

1151 JMTPROC4(1)
1152          JMTPROC4
1153 JREADBACK(20)
1154          A = 1
1155          $ = 4
1156          $E18
1157          $ = 13
1158      U,  $ - MA[-6],  Z
1159      N,  SUBCD(:PSE19)
1160          F = :MA[-11]
1161          SUBCD(:PSE20)
1162          $ = 16
1163          $E1
1164          DOS(MA[-11])
1165          DO(ERV)
1166      U,  $ = E,  Z
1167      N,  SUBCD(:PSE88)
1168          $UBC(:CREADBACK)
1169          $ = JMTREWRRR
1170          GBILL + $, R
```

```
1171          $E115
1172          DO(MD[ - 1])
1173          $E71

1174 CREADBACK1(42)
1175          G = JCURUNIT
1176          $ = :MG[3]
1177          SUBCD(:UKILLSEGR)
1178          A = MG[19], Z                    " LAST ELEMENT = 0 ?
1179      Y,  MG[17] = - B
1180      Y,  JUMP(33)
1181          A = MG[21], P                    " READFORW ?
1182      N,  JUMP(2)
1183          CWEEN
1184          JSROELBACKW
1185          $ = :MG[3]
1186          SUBCD(:UHOLYRMSEG)
1187          $ = MS
1188          $ = - MS[2]
1189          $ - 512
1190          $ + UCON1                        " VAKSPECIFICATIE
1191          A = 512
1192          CMTR
1193          CWEEN
1194          JREADB
1195          $ = - JLABELMT[4]
1196          CMTV
1197          $ '*' UCON2
1198          $ = MS
1199      U,  $ '*' '60', Z
1200      N,  JUMP(3)
1201          A = 416
1202          $E33
1203          JALARM
1204          RUS(9)
1205          $ '*' '7777'
1206          MG[15] = S                       " FILL BLOCKLENGTH
1207          MG[21] = - B                     " READFORW := EALSE
1208          A = 1
1209          MG[10] = A
1210          $ = :MG[3]
1211          SUBCD(:URROEANATION)
1212          A = 509
1213          MG[16] = A                       " SET POS
1214          A = - 1
1215          $E33
1216          CREADTAPE1[12]

1217 JMTMES00C(7)
1218          '030 036 035'                    " O U T
1219          '135 030 017'                    "   O F
1220          '135 014 030'                    "   C O
1221          '027 035 033'                    " N T R
1222          '030 025 135'                    " O L
1223          '036 027 022'                    " U N I
1224          '035 135 776'                    " T   END

1225 JMTMESREV(9)
1226          '040 012 022'                    " W A I
1227          '035 022 027'                    " T I N
```

end of mag.tape procedures

```
1228          '020 135 017'                    " G   F
1229          '030 033 135'                    " O R
1230          '033 016 040'                    " R E W
1231          '022 027 015'                    " I N D
1232          '135 036 027'                    "   U N
1233          '022 035 135'                    " I T
1234          '776 000 000'                    " END

1235 JLOADREEL(4)
1236          '025 030 012'                    " L O A
1237          '015 135 033'                    " D   R
1238          '016 016 025'                    " E E L
1239          '135 776 000'                    "   END

1240 JONUNIT(3)
1241          '030 027 135'                    " O N
1242          '036 027 022'                    " U N I
1243          '035 135 776'                    " T   END

1244 JMTNBKMES(4)
1245          '035 012 031'                    " T A P
1246          '016 034 135'                    " E S
1247          '027 013 024'                    " N B K
1248          '130 776 000'                    " . END

1249 JINVM1(7)
1250          '036 027 022'                    " U N I
1251          '035 135 015'                    " T   D
1252          '022 034 014'                    " I S C
1253          '030 027 027'                    " O N N
1254          '016 014 035'                    " E C T
1255          '016 015 130'                    " E D .
1256          '776 000 000'                    " END

1257 JINVM3(6)
1258          '036 027 022'                    " U N I
1259          '035 135 033'                    " T   R
1260          '016 014 030'                    " E C O
1261          '027 027 016'                    " N N E
1262          '014 035 016'                    " C T E
1263          '015 130 776'                    " D . END

1264 JEINDMES(11)
1265          '167 026 012'                    " NLCR M A
1266          '020 027 016'                    " G N E
1267          '016 035 013'                    " E T B
1268          '012 027 015'                    " A N D
1269          '012 017 034'                    " A F S
1270          '025 036 022'                    " L U I
1271          '035 022 027'                    " T I N
1272          '020 135 026'                    " G   M
1273          '022 034 025'                    " I S L
1274          '036 024 035'                    " U K T
1275          '130 167 776'                    " . NLCR END

1276 UE249(22)
1277          '002 004 011'                    " 2 4 9
1278          '130 135 015'                    "     D
1279          '042 027 012'                    " Y N A
1280          '026 022 034'                    " M I S
```

```
1281          '014 021 135'                    " C H
1282          '020 016 027'                    " G E N
1283          '016 034 035'                    " E S T
1284          '016 135 012'                    " E   A
1285          '012 027 033'                    " A N R
1286          '030 016 031'                    " O E P
1287          '135 037 012'                    "   V A
1288          '027 135 063'                    " N   O
1289          '064 051 062'                    " P E N
1290          '070 045 064'                    " T A P
1291          '051 135 037'                    " E   V
1292          '030 030 033'                    " O O R
1293          '135 016 016'                    "   E E
1294          '027 135 043'                    " N   Z
1295          '016 025 017'                    " E L F
1296          '015 016 135'                    " D E
1297          '033 016 016'                    " R E E
1298          '025 776 000'                    " L END

1299 UF250(16)
1300          '002 005 000'                    " 2 5 0
1301          '130 135 012'                    "     A
1302          '012 027 035'                    " A N T
1303          '012 025 135'                    " A L
1304          '013 016 034'                    " B E S
1305          '014 021 022'                    " C H I
1306          '024 013 012'                    " K B A
1307          '033 016 135'                    " R E
1308          '035 012 031'                    " T A P
1309          '016 036 027'                    " E U N
1310          '022 035 034'                    " I T S
1311          '135 030 037'                    "   O V
1312          '016 033 034'                    " E R S
1313          '014 021 033'                    " C H R
1314          '016 015 016'                    " E D E
1315          '027 776 000'                    " N END

1316 UF251(13)
1317          '002 005 001'                    " 2 5 1
1318          '130 135 033'                    "     R
1319          '016 016 025'                    " E E L
1320          '013 025 030'                    " B L O
1321          '024 135 030'                    " K   O
1322          '017 135 025'                    " F   L
1323          '012 013 016'                    " A B E
1324          '025 013 025'                    " L B L
1325          '030 024 135'                    " O K
1326          '026 036 035'                    " M U T
1327          '022 025 012'                    " I L A
1328          '035 016 015'                    " T E D
1329          '776 000 000'                    " END

1330 UF252(14)
1331          '002 005 002'                    " 2 5 2
1332          '130 135 031'                    "     P
1333          '012 033 012'                    " A R A
1334          '026 016 035'                    " M E T
1335          '016 033 135'                    " E R
1336          '027 012 026'                    " N A M
1337          '016 135 022'                    " E   I
```

dynamic errors
occurring in tape procedures

```
1338          '027 135 063'                  " N   Q
1339          '064 051 062'                  " R E N
1340          '070 045 064'                  " T A R
1341          '051 135 035'                  " E   T
1342          '016 135 025'                  " E   L
1343          '012 027 020'                  " A N G
1344          '776 000 000'                  " END

1345 UE253(19)
1346          '002 005 003'                  " 2 5 3
1347          '130 135 022'                  "     I
1348          '027 135 063'                  " N   Q
1349          '064 051 062'                  " R E N
1350          '070 045 064'                  " T A R
1351          '051 135 027'                  " E   N
1352          '012 026 016'                  " A M E
1353          '135 027 022'                  "   N I
1354          '016 035 135'                  " E T
1355          '020 016 025'                  " G E L
1356          '022 023 024'                  " I J K
1357          '135 012 012'                  "   A A
1358          '027 135 025'                  " N   L
1359          '012 013 016'                  " A B E
1360          '025 135 037'                  " L   V
1361          '012 027 135'                  " A N
1362          '015 016 135'                  " D E
1363          '033 016 016'                  " R E E
1364          '025 776 000'                  " L END

1365 UE254(11)
1366          '002 005 004'                  " 2 5 4
1367          '130 135 063'                  "     Q
1368          '064 051 062'                  " R E N
1369          '070 045 064'                  " T A R
1370          '051 135 027'                  " E   N
1371          '022 016 035'                  " I E T
1372          '135 012 012'                  "   A A
1373          '027 020 016'                  " N G E
1374          '033 030 016'                  " R O E
1375          '031 016 027'                  " P E N
1376          '776 000 000'                  " END

1377 UE255(25)
1378          '002 005 005'                  " 2 5 5
1379          '130 135 022'                  "     I
1380          '027 135 012'                  " N   A
1381          '012 027 033'                  " A N R
1382          '030 016 031'                  " O E P
1383          '135 037 012'                  "   V A
1384          '027 135 073'                  " N   W
1385          '066 055 070'                  " R I T
1386          '051 101 135'                  " E -
1387          '030 017 135'                  " O F
1388          '066 051 045'                  " R E A
1389          '050 064 063'                  " D R O
1390          '067 055 070'                  " S I T
1391          '055 063 062'                  " I O N
1392          '135 021 016'                  "   H E
1393          '016 017 035'                  " E F T
1394          '135 016 025'                  "   E L
```

```
1395          '016 026 016'                    " E M E
1396          '027 035 135'                    " N T
1397          '027 016 020'                    " N E G
1398          '012 035 022'                    " A T I
1399          '016 037 016'                    " E V E
1400          '135 040 012'                    "   W A
1401          '012 033 015'                    " A R D
1402          '016 776 000'                    " E END

1403 UR256(24)
1404          '002 005 006'                    " 2 5 6
1405          '130 135 022'                    "     I
1406          '027 135 012'                    " N   A
1407          '012 027 033'                    " A N R
1408          '030 016 031'                    " O E P
1409          '135 037 012'                    "   V A
1410          '027 135 073'                    " N   W
1411          '066 055 070'                    " R I T
1412          '051 101 135'                    " E -
1413          '030 017 135'                    " O F
1414          '066 051 045'                    " R E A
1415          '050 064 063'                    " D P Q
1416          '067 055 070'                    " S I T
1417          '055 063 062'                    " I Q N
1418          '135 021 016'                    "   H E
1419          '016 017 035'                    " E F T
1420          '135 020 033'                    "   G R
1421          '030 036 031'                    " O U P
1422          '135 027 016'                    "   N E
1423          '020 012 035'                    " G A T
1424          '022 016 037'                    " I E V
1425          '016 135 040'                    " E   W
1426          '012 012 033'                    " A A R
1427          '015 016 776'                    " D E END

1428 UE257(19)
1429          '002 005 007'                    " 2 5 7
1430          '130 135 012'                    "     A
1431          '012 027 033'                    " A N R
1432          '030 016 031'                    " O E P
1433          '135 037 012'                    "   V A
1434          '027 135 073'                    " N   W
1435          '066 055 070'                    " R I T
1436          '051 070 045'                    " E T A
1437          '064 051 135'                    " R E
1438          '135 027 012'                    "   N A
1439          '135 022 027'                    "   I N
1440          '014 030 033'                    " C O R
1441          '033 016 014'                    " R E C
1442          '035 016 135'                    " T E
1443          '031 030 034'                    " P O S
1444          '022 035 022'                    " I T I
1445          '030 027 016'                    " O N E
1446          '033 022 027'                    " R I N
1447          '020 776 000'                    " G END

1448 UE258(16)
1449          '002 005 010'                    " 2 5 8
1450          '130 135 012'                    "     A
1451          '012 027 033'                    " A N R
```

```
1452          '030 016 031'                  " O E P
1453          '135 037 012'                  "   V A
1454          '027 135 073'                  " N   W
1455          '066 055 070'                  " R I T
1456          '051 070 045'                  " E T A
1457          '064 051 135'                  " R E
1458          '037 030 030'                  " V O O
1459          '033 135 013'                  " R   B
1460          '016 034 014'                  " E S C
1461          '021 016 033'                  " H E R
1462          '026 015 016'                  " M D E
1463          '135 035 012'                  "   T A
1464          '031 016 776'                  " P E END

1465 VE259(12)
1466          '002 005 011'                  " 2 5 9
1467          '130 135 016'                  "       E
1468          '022 027 015'                  " I N D
1469          '016 135 026'                  " E   M
1470          '012 020 027'                  " A G N
1471          '016 016 035'                  " E E T
1472          '013 012 027'                  " B A N
1473          '015 135 020'                  " D   G
1474          '016 031 012'                  " E P A
1475          '034 034 016'                  " S S E
1476          '016 033 015'                  " E R D
1477          '776 000 000'                  " END

1478 VE260(20)
1479          '002 006 000'                  " 2 6 0
1480          '130 135 012'                  "       A
1481          '012 027 033'                  " A N R
1482          '030 016 031'                  " O E P
1483          '135 037 012'                  "   V A
1484          '027 135 025'                  " N   L
1485          '016 016 034'                  " E E S
1486          '031 033 030'                  " P R O
1487          '014 016 015'                  " C E D
1488          '036 033 016'                  " U R E
1489          '135 027 012'                  "   N A
1490          '135 022 027'                  "   I N
1491          '014 030 033'                  " C O R
1492          '033 016 014'                  " R E C
1493          '035 016 135'                  " T E
1494          '031 030 034'                  " P O S
1495          '022 035 022'                  " I T I
1496          '030 027 016'                  " O N E
1497          '033 022 027'                  " R I N
1498          '020 776 000'                  " G END

1499 VE261(16)
1500          '002 006 001'                  " 2 6 1
1501          '130 135 022'                  "       I
1502          '027 135 025'                  " N   L
1503          '016 016 034'                  " E E S
1504          '031 033 030'                  " P R   O
1505          '014 135 020'                  " C   G
1506          '033 016 027'                  " R E N
1507          '034 135 037'                  " S   V
1508          '012 027 135'                  " A N
```

```
1509          '015 016 135'                    " D E
1510          '020 033 030'                    " G R O
1511          '016 031 135'                    " E P
1512          '020 016 031'                    " G E P
1513          '012 034 034'                    " A S S
1514          '016 016 033'                    " E E R
1515          '015 776 000'                    " D END

1516 UF262(16)
1517          '002 006 002'                    " 2 6 2
1518          '130 135 031'                    "     P
1519          '012 033 022'                    " A R I
1520          '035 016 022'                    " T E I
1521          '035 034 017'                    " T S F
1522          '030 036 035'                    " O U T
1523          '135 013 022'                    "   B I
1524          '023 135 034'                    " J   S
1525          '014 021 033'                    " C H R
1526          '022 023 037'                    " I J V
1527          '016 027 135'                    " E N
1528          '030 031 135'                    " O P
1529          '026 012 020'                    " M A G
1530          '027 016 016'                    " N E E
1531          '035 013 012'                    " T B A
1532          '027 015 776'                    " N D END

1533 UF263(15)
1534          '002 006 003'                    " 2 6 3
1535          '130 135 031'                    "     P
1536          '012 033 022'                    " A R I
1537          '035 016 022'                    " T E I
1538          '035 034 017'                    " T S F
1539          '030 036 035'                    " O U T
1540          '135 013 022'                    "   B I
1541          '023 135 025'                    " J   L
1542          '016 043 016'                    " E Z E
1543          '027 135 037'                    " N   V
1544          '012 027 135'                    " A N
1545          '026 012 020'                    " M A G
1546          '027 016 016'                    " N E E
1547          '035 013 012'                    " T B A
1548          '027 015 776'                    " N D END

1549 UF264(19)
1550          '002 006 004'                    " 2 6 4
1551          '130 135 034'                    "     S
1552          '042 027 035'                    " Y N T
1553          '012 014 035'                    " A C T
1554          '022 034 014'                    " I S C
1555          '021 016 135'                    " H E
1556          '034 035 033'                    " S T R
1557          '036 014 035'                    " U C T
1558          '036 036 033'                    " U U R
1559          '135 037 012'                    "   V A
1560          '027 135 015'                    " N   D
1561          '016 135 026'                    " E   M
1562          '012 020 027'                    " A G N
1563          '016 016 035'                    " E E T
1564          '013 012 027'                    " B A N
1565          '015 135 022'                    " D   I
```

```
1566          '027 014 030'                    " N C O
1567          '033 033 016'                    " R R E
1568          '014 035 776'                    " C T END

1569 WR265(21)
1570          '002 006 005'                    " 2 6 5
1571          '130 135 012'                    "     A
1572          '012 027 033'                    " A N R
1573          '030 016 031'                    " O E P
1574          '135 037 012'                    "   V A
1575          '027 135 016'                    " N   E
1576          '016 027 135'                    " E N
1577          '035 012 031'                    " T A P
1578          '016 135 031'                    " E   P
1579          '033 030 014'                    " R O C
1580          '016 015 036'                    " E D U
1581          '033 016 135'                    " R E
1582          '043 030 027'                    " Z O N
1583          '015 016 033'                    " D E R
1584          '135 037 030'                    "   V O
1585          '030 033 012'                    " O R A
1586          '017 020 012'                    " F G A
1587          '012 027 015'                    " A N D
1588          '016 135 014'                    " E   C
1589          '025 012 022'                    " L A I
1590          '026 776 000'                    " M END

1591 WR266(13)
1592          '002 006 006'                    " 2 6 6
1593          '130 135 034'                    "     S
1594          '035 030 033'                    " T O R
1595          '022 027 020'                    " I N G
1596          '135 022 027'                    "   I N
1597          '135 026 012'                    "   M A
1598          '020 027 016'                    " G N E
1599          '016 035 013'                    " E T B
1600          '012 027 015'                    " A N D
1601          '012 031 031'                    " A P P
1602          '012 033 012'                    " A R A
1603          '035 036 036'                    " T U U
1604          '033 776 000'                    " R END

1605 JMTVERSLAGKOP(26)
1606          '167 167 167'                    " NLCR NLCR NLCR
1607          '135 135 135'                    "
1608          '135 135 135'                    "
1609          '135 033 016'                    "   R E
1610          '016 025 135'                    " E L
1611          '027 033 135'                    " N R
1612          '135 135 135'                    "
1613          '135 025 012'                    "   L A
1614          '012 035 034'                    " A T S
1615          '035 016 135'                    " T E
1616          '020 033 030'                    " G R O
1617          '016 031 135'                    " E P
1618          '027 033 135'                    " N R
1619          '135 135 037'                    "     V
1620          '030 025 020'                    " O L G
1621          '016 027 015'                    " E N D
1622          '135 016 025'                    "   E L
```

```
1623          '016 026 016'                    " E M E
1624          '027 035 135'                    " N T
1625          '027 033 135'                    " N R
1626          '135 135 025'                    "     L
1627          '012 012 035'                    " A A T
1628          '034 035 016'                    " S T E
1629          '135 026 030'                    "   M O
1630          '015 036 034'                    " D U S
1631          '167 167 776'                    " NLCR NLCR NLCR

1632 JMTVERSLAGMES(19)
1633          '167 167 031'                    " NLCR NLCR P
1634          '033 030 020'                    " R O G
1635          '033 012 026'                    " R A M
1636          '026 012 135'                    " M A
1637          '013 016 016'                    " B E E
1638          '022 027 015'                    " I N D
1639          '022 020 015'                    " I G D
1640          '135 035 020'                    "   T G
1641          '037 135 034'                    " V   S
1642          '035 030 033'                    " T O R
1643          '022 027 020'                    " I N G
1644          '135 026 012'                    "   M A
1645          '020 027 016'                    " G N E
1646          '016 035 013'                    " E T B
1647          '012 027 015'                    " A N D
1648          '012 031 031'                    " A P P
1649          '012 033 012'                    " A R A
1650          '035 036 036'                    " T U U
1651          '033 130 776'                    " R END

1652 JMTMODELEES(4)
1653          '135 135 135'                    "
1654          '135 135 025'                    "     L
1655          '016 016 034'                    " E SE S
1656          '167 776 000'                    " NLCR END

1657 JMTMODESCHRIJF(4)
1658          '135 135 034'                    "     S
1659          '014 021 033'                    " G H R
1660          '022 023 017'                    " I J F
1661          '167 776 000'                    " NLCR END

1662 JMTERRORMES(7)
1663          '033 016 016'                    " R E E
1664          '025 135 016'                    " L   E
1665          '033 033 030'                    " R R O
1666          '033 034 135'                    " R S
1667          '035 030 035'                    " T O T
1668          '012 025 135'                    " A L
1669          '135 135 776'                    "     END

1670 UR267(12)
1671          '002 006 007'                    " 2 6 7
1672          '130 135 031'                    "     P
1673          '012 033 012'                    " A R A
1674          '026 016 035'                    " M E T
1675          '016 033 135'                    " E R
1676          '033 016 016'                    " R E E
1677          '025 135 022'                    " L   I
```

```
1678          '027 135 063'                    " N   Q
1679          '064 051 062'                    " R E N
1680          '070 045 064'                    " T A R
1681          '051 135 111'                    " E   S
1682          '000 776 000'                    " 0 END

1683 KARAKTERUIT(82)
1684          A =1
1685          S=4
1686          SE18
1687          S=MA[-6]
1688 U,       S-13,Z                           "IS HET AANTAL PARAMETERS
1689 N,       SUBCD(:RSE19)                    "GELIJK AAN 1?
1690          F=:MA[-11]                       "IS DE PARAMETER VAN HET
1691          SUBCD(:RSE20)                    "GOEDE TYPE
1692          S=16
1693          SE1                              "RESERVEER STAPELRUIMTE
1694          DOS(MA[-11])                     "BEREKEN
1695          DO(ERV)                          "DE PARAMETER
1696          S=M[57],Z                        "MAAK DE PARAMETERWAARDE
1697 N,       SUBCD(:RSE88)                    "VAN HET TYPE INTEGER
1698          MA[-11]=G
1699          UDEAF
1700          S=:M0[0]
1701     U,   S - CONSOLECLAIMED, Z                    " IS IN DE KOP VAN HET PROGRAMMA
1702 N,       A=268                            "OM DE CONSOLE GEVRAAGD?
1703 N,       SUBCD(:UDYNERR)                  "ZO NIET, FOUTMELDING
1704          SUBCD(:UIMMORTAL)
1705          G = MA[-11],R                    " ALS DE PARAMETERWAARDE NIET
1706     N,   F = - F, Z                       " OVEREENKOMT MET DE INTERNE
1707     N,   F = 128                          " CODE VAN EEN KARAKTER, LEVER
1708          A = G                            " DAN # AF
1709          F - 128, R
1710     Y,   A = 128
1711 U,       A-36,R                           "BRENG DE LETTERS TERUG
1712 Y,       A-27                             "OP EEN TYPE
1713          F = :MT[18]                      " BEGINADRES OMCODERINGSTABEL
1714          RWAS(26)
1715          DIVAS(3),Z                       "A:=POSITIE VAN DE CODE IN TABELWOORD
1716          S+G                              "ADRES VAN TABELWOORD
1717          S=M$                             "TABELWOORD IN S
1718 N,       RCS(9)
1719 N,       A-1,Z                            "ZOEK JUISTE POSITIE IN TABELWOORD
1720 N,       RCS(9)
1721          S'*'511                          "BEHOUD 9 BITS IN S
1722          A=S                              "SEMI HARDWARE REPRESENTATIE IN A
1723          UKLACH = B                       " MAAK KLEUR ZWART
1724          UGAPR=B                          "GA PRINTEN:=IBUE, DWZ IEDER KARAKTER METEEN PRINTEN
1725          SUBCD(:MT[98])                   " GA NAAR EIGENLIJKE PRINTROUTINE
1726          SUBCD(:UMORTAL)
1727          DO(MD[-1])
1728          S=25                             "100 MILISECONDEN
1729          GBILL+S,R                        "MAAK DE REKENING OP
1730          SE115
1731          SE71
1732          '071     075     055'
1733          '041     052     060'
1734          '054     074     065'
1735          '023     030     043'
1736          '020     022     016'
```

more procedures for the teletype terminal

```
1737          '005    013     026'
1738          '036    032     014'
1739          '006    007     011'          "OMCODERINGSTABEL VAN
1740          '035    015     003'          "INTERNE REPRESENTATIE
1741          '001    024     012'          "NAAR SEMI-HARDWARE REPRESENTATIE
1742          '031    017     034'
1743          '021    025     027'
1744          '070    061     000'
1745          '000    067     063'
1746          '000    057     000'
1747          '000    000     000'
1748          '000    000     000'
1749          '000    000     000'
1750          '000    000     000'
1751          '000    000     000'
1752          '045    047     046'
1753          '000    072     056'
1754          '000    000     004'
1755          '076    000     000'
1756          '053    066     051'
1757          '000    000     000'
1758          '000    000     000'
1759          '000    000     000'
1760          '000    000     000'
1761          '000    000     000'
1762          '002    000     000'
1763          '000    000     064'
1764          '000    062     040'
1765          '000    000     000'
1766          KARAKTERIN(160)
1767          A=1
1768          S=4
1769          SE18
1770          S=MA[-6]
1771 U,       S-9,Z                         "AANTAL PARAMETERS GELIJK AAN 0?
1772 N,       SUBCD(:RSE19)
1773          UDEAF
1774          S=:M0[0]
1775          S-CONSOLECLAIMED,Z            "IS IN DE KOP VAN HET PROGRAMMA
1776      N,  JUMP(-74)                     " OM DE CONSOLE GEVRAAGD ?
1777          SUBCD(:MT[145])
1778          SUBCD(:MT[103])               "MAAK IET NUL
1779          A=UARCKB
1780          SUBCD(:MT[107])               "CASE?
1781          SUBCD(:MT[113])               "AFSLUITING
1782          R=:UIETRC                     "<EIGENLIJKE LEESROUTINE>
1783          URH                           "R(IETRC)
1784          A=UARCKB[1]                   "A:=AR1
1785          A'*' -UCON2, Z                " IETRC = 0 ?
1786 Y,       JUMP(6)
1787          SUBC(:MT[94])                 "MAAK IET NUL
1788          A=UARCKB[0],R                 "AR0 ≥ +0?
1789 N,       JUMP(-8)
1790          SUBC(:MT[97])                 "CASE?
1791          SUBC(:MT[103])                "AFSLUITING
1792          JUMP(-11)
1793          A=UARCKB[0],R
1794 N,       JUMP(-13)
1795 U,       A-31,R                        "A > 31, DWZ ONGELDIG KARAKTER
1796 N,       SUBC(:MT[91])                 "CASE?
```

```
1797 Y,       JUMP(-7)
1798 U,       A-8,Z                    "VERWERK KARAKTER
1799 Y,       A=2                      "MAAK VAN NR EEN TW
1800 U,       A-2,Z                    "TW?
1801 Y,       UTWNR=A                  "BOODSCHAP HERROEPEN DOOR OPERATEUR := IBUE
1802 Y,       JUMP(2)
1803 U,       A-4,Z                    "SPATIE?
1804 N,       A+CASERC                 "MAAK SEMI-HARDWARE REPRESENTATIE
1805          UGARR=B                  "GA PRINTEN := IBUE
1806          UKLACH = - B             " KLEUR WORDT ROOD
1807          SUBC(:MT[17])            "TERUGPRINTEN
1808          MC=A                     "RED A VOOR AFSLUITING
1809          SUBC(:MT[85])            "AFSLUITEN
1810          E = :MT[91]                      " BEGINADRES OMCODERINGSTABEL
1811          S=MC[-1]
1812          A=0                      "GA PLAATS IN DE TABEL ZOEKEN
1813          DIVAS(3),Z               "A:=POSITIE VAN CODE IN TABELWOORD
1814          S+G                      "ADRES VAN TABELWOORD
1815          S=MS                     "TABELWOORD IN S
1816 N,       RCS(9)
1817 N,       A-1,Z                    "PAK HET GOEDE STUK
1818 N,       RCS(9)
1819          S'*'511                  "BEHOUD 9 BITS IN S
1820          E = :EORSEM
1821          UVWIT
1822          SUBCD(:UMORTAL)
1823          G = S
1824          JUMP( -97)
1825          SUBCD(:UREDDEN)          "<EIGENLIJKE PRINTROUTINE>
1826 U,       A-2,Z                    "IS A = TWNR
1827 Y,       JUMP(28)
1828 U,       A-31,R                   "IS CASE = CY
1829 Y,       A-32                     "MAAK HARDWARE REPRESENTATIE
1830          S=CASEDWTT,E             "CASEDWTT=CASECHAR?
1831 N,       CASEDWTT=-S              "CASEDWTT:=CASECHAR
1832 N,       JUMP(3)
1833          S=UKLACH,E
1834          S=UKLDW,E                "KLEURDWTT=KLEURCHAR
1835 Y,       JUMP(9)
1836          S=UKLACH
1837          UKLDW=S,R                "KLEURDWTT:=KLEURCHAR, ZWART?
1838 Y,       E=32
1839 N,       E=33
1840          S=CASEDWTT,R
1841 N,       E+4
1842          S=1
1843          PLUSS(UMAGADRES)         "VW:=VW+1
1844          MS[-1]=G                 "ZET CASE/KLEUR IN MAGAZIJN
1845          A+0,Z
1846 Y,       A=0                      "MAAK VAN 0 EEN +0
1847          S=1
1848          PLUSS(UMAGADRES)
1849          MS[-1]=A                 "ZET KARAKTER IN MAGAZIJN
1850          A-8,Z                    "WAS KARAKTER NR
1851 N,       A=1                      "HWP=HWP+1
1852          PLUSA(UHWP)
1853          A-UMAXWP,Z               "REGELBREEDTE BEREIKT?
1854 N,       JUMP(8)
1855          A=2
1856          S=2
```

```
1857          PLUSS(UMAGADRES)                  "VW:=VW+2
1858          MS[-2]=A                          "IN MAGAZIJN TW
1859          A=8
1860          MS[-1]=A                          "IN MAGAZIJN NR
1861          A=0
1862          UHVR=A
1863          S-UMAXMAG,R                       "MAGAZIJN VOL?
1864 N,       A=UGAPR,R                         "GA PRINTEN=IRUE?
1865 N,       JUMR(15)
1866          S + UMAGLE[- 4]                   " S := AANTAL GEVULDE PLAATSEN
1867          A=TTLABEL[2]
1868          LCA(9)
1869          RUSA(9)
1870          TTLABEL[2]=A                      "CODEWOORD IN LABEL[1] IN ORDE
1871          A = :TTLABEL[1]
1872          UARCRB[0]=A                       "ARO(TT) := (:LABEL[0])
1873          A=UCON1                           " 2↑16
1874          UARCRB[2]=A                       "AF(TT):=1
1875          DO(UIETTT[7])                     "ZET AF(TT) IRUE
1876          F=:UIETTT[0]
1877          URH
1878          S=:UMAGAZYN
1879          UMAGADRES=S                       "HERINITIALISEER MAGAZIJNWIJNER
1880          UGAPR=-B                          "GA PRINTEN := FALSE
1881          GOTO(:UHERSTELTERUG)              "EINDE PRINTROUTINE
1882          S=2                               "<MAAK IFT NUL>
1883          UARCKB[1]=S                       "IFTRC:=0
1884          DO(UIFTRC[4])                     "IFRC:=FALSE
1885 U,       S-UARCKB[1],Z                     "IF IFTRC=0 IHEN ...
1886 N,       JUMR(-4)
1887          GOTOR(MC[-1])
1888 U,       A-27,Z                                  "<CASE> IS AANSLAG = CY
1889 Y,       A=32
1890 Y,       JUMR(2)                                 "IS AANSLAG = LF
1891 U,       A-31,Z
1892 Y,       A=0
1893 Y,       CASERC=A
1894          GOTO(MC[-1])
1895          S=1
1896                                  "<AFSLUITING>
1897          UARCKB[0]=-S                            "AR0 = -1
1898          DO(UIFTRC[7])                           "AF := IRUE
1899          A=UARCKB[3]                             "A:=AR3
1900          SUBC(:MT[-12])                          "ALS HET EEN CASE IS DAN BIJHOUDEN
1901          UARCKB[3]=-S                            "AR3:=-1
1902          GOTOR(MC[-1])
1903          '167 070 175'
1904          '054     135     063'
1905          '000     061     062'
1906          '053     066     060'
1907          '047     064     055'
1908          '076     051     072'
1909          '067     046     050'
1910          '074     052     075'
1911          '056     073     045'
1912          '065     071     000'
1913          '173     000     057'                   "OMCODERINGSTABEL VAN
1914          '011     000     005'                   "SEMI-HARDWARE REPRESENTATIE
1915          '127     131     000'                   "NAAR INTERNE REPRESENTATIE
1916          '143     000     130'
```

```
1917          '010    145     004'
1918          '106    132     000'
1919          '174    100     003'
1920          '006    170     102'
1921          '101    103     144'
1922          '000    133     002'
1923          '142 001 007'
1924          SUBCD(:U:MMORTAL)
1925          F = :FORSEM
1926          UPBOOD
1927          GOTOR(MC(- 11)
```

end of system