VALLEE

NIC 11863

# SCENARIOS

## for using the

## ARPANET

## at the

## INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION

Washington, D.C.
October 24–26, 1972

ARPA Network Information Center
Stanford Research Institute
Menlo Park, California 94025

Tue. 11:30 - 3
6:30 - 10

Matt Bernstein

Wed 8:30 - 11:30
3 - 6:30

Thu 11:30 - 3

NIC 11863

# SCENARIOS

## for using the

## ARPANET

## at the

## INTERNATIONAL CONFERENCE ON COMPUTER COMMUNICATION

Washington, D.C.
October 24–26, 1972

ARPA Network Information Center
Stanford Research Institute
Menlo Park, California 94025

# SCENARIOS FOR USING THE ARPANET AT THE ICCC

We intend that the following scenarios be used by individuals to browse the ARPA Computer Network (ARPANET) in its current early stage of development and thereby to introduce themselves to some possibilities in computer communication. The scenarios include only a few of the existing ARPANET resources. They were chosen for this booklet (somewhat haphazardly) to exhibit variety and sophistication, while retaining simplicity.

The scenarios are by no means complete or perfect. We have tried to make them accurate, but are certain that they contain errors. The scenarios are, therefore, only one kind of tool for experiencing computer communication.

We assume that you will attend the various showings of film and videotape, pay close attention at the several scheduled demonstrations of specific resources, approach the ARPANET aggressively yourself using these scenarios, and unhesitatingly call upon the ICCC Special Project People for the advice and encouragement you are sure to need.

The account numbers and passwords provided in these scenarios were generated specifically for the ICCC. It is hoped that some of them will remain available after the ICCC for continued browsing. It is expected that the scenarios will have long term usefulness.

The idea for a scenario booklet came out of preparations for a system programmers workshop at MIT in 1971. Abhay Bhushan collected the first set of ARPANET scenarios at MIT and has contributed much to the scenarios included here. Too many people to name have worked on developing scenarios for the ICCC and providing computing resources. Mention should be made of the Network Information Center at the Stanford Research Institute where this booklet was collected, edited, validated, and produced. We specifically acknowledge the help of Dick Watson, Marilyn Auerbach, Linda Lane, Barbara Row, and Kay Byrd in editing and production.

Bob Metcalfe
Xerox PARC

October 24, 1972

ICCC Scenario Conventions

To make browsing easier, we have chosen conventions with which to specify console sessions and attempted to use them uniformly throughout this booklet.

The scenarios are written for use from an ARPANET TIP. Each scenario begins with a series of TIP commands. These include the setting of certain modes (e.g., line-at-a-time) to suit the serving HOST. The scenarios do not include the several TIP commands used to establish terminal-dependent parameters (e.g., extra-padding after carriage returns for timing in fast terminals). Such terminal-dependent commands should be entered at the start of each console session as specified by parameter sheets supplied with each terminal.

In the scenarios, that which a user is expected to type is underlined to set it off from computer type-out, general instructions, and italicized comments. For example:

The computer asks whether it should proceed and the user responds with "yes" followed by carriage return:

PROCEED?

*"Should I proceed?", the computer asks.*

yes [CR]

*"yes", the user answers (in lower case).*

We have tried to help the user over common trouble spots by paying attention to whether he should type upper or lower case alphabetics and by including clearly marked special characters where EMPHASIS is warranted:

LOGIN PLEASE

login [SP] iccc [SP] CNet [CR]

*Note the case shift at "CNet" and the spaces (SP) and carriage return (CR).*

Special characters used include:

CR = Carriage return, RET, Return, CR
LF = Linefeed, Newline, LF
ESC = Escape, Altmode, ESC
SP = Space, SP

Because of their frequent occurrence, we should state that teletype "control characters" are denoted by up-arrow followed by the specifying alphabetic. For example, control-c, written "↑c", is typed by holding down the CTRL or CONTROL key on a teletype-like device while striking the "c" key.

If you have difficulty following any or all of the scenarios, please ask an ICCC Special Project person for help, rather than stew in your own juices.

TABLE OF CONTENTS

Scenarios for Using the ARPANET at the ICCC



SCENARIOS BY (APPROXIMATE) CATEGORY

MIT-DMCG ITS PDP-10 === HOST #70

The MIT Project MAC Dynamic Modelling and Computer Graphics (DMCG) PDP-10 runs the ITS time-sharing system developed at the MIT Artificial Intelligence Laboratory.

ITS prefers to do its own echoing, a character at a time. Its attention getting character is control-z ("↑Z"). Typing DEL or RUBOUT will generally delete the last character typed on input. Control-g will generally abort commands. To suppress output, type control-s. At command level, upper and lower case alphabetics are treated alike.

1. To set TIP parameters and connect to MIT-DMCG:

```
@r [LF]                    Reset the TIP. Terminal-dependent setup should be done here.
@e [SP] r [LF]             "Echo remote" is preferred by DMCG ITS.
@L [SP] 70 [LF]            Cause TIP "LOGGER" to connect to DMCG, HOST #70.

LOGGER
T R OPEN
MIT PROJECT MAC DMCG PDP-10.
......                     Various system messages appear here and can be suppressed
                           with control-s ("↑S").
```

2. To login, type:

```
; log [SP] iccXXX [CR]     "XXX" should be your initials to avoid LOGIN conflicts.
                           Note that ";" is the ITS MONITOR prompt character.
```

3. To see who is using the system:

```
; who [CR]
TTY UNAME  JNAME  CORE TOTAL IDX
T07 SYS    SYS    035  035   01
D11 ++++++ UNSPOO 001  007   24
T13 PDL    T      007  008   03
T16 ++++++ HACTRN 001  001   10
T17 AKB    NETWRK 006  007   05
T21 PMA    HACTRN 001  003   06
T31 ICCXXX HACTRN 001  001   23
DSN PJ     SPLASH 004  007   07
FREE CORE  117/208
```

4. To send mail to another ICCC DMCG ITS user, type:

```
; mail [SP] iccakb [SP] hello [CR]
```

*Note that in the example, the mail is being sent to ICCC user "akb". Because ICCAKB has no file directory at DMCG, his mail is put in a common file directory (COM).*

*This method of sending mail allows one-line messages only.*

```
USER HAS NO DIRECTORY, MAIL GOING TO COM.
```

5. To read a user's mail, type:

```
; mail [SP] iccakb [CR]
```

*You are requesting that all of a user's pending mail be typed on your console. This collection of mail will also be typed out when the user in question logs in next time. You might try sending mail to yourself, as above, to get mail the next time you log into DMCG ITS during the ICCC.*

```
FROM ICCXXX 09/20/72 20:07:35
HELLO.
```

6. To use the DMCG ITS ARPANET NETWRK subsystem, type:

```
; netwrk [CR]
\NETWRK.202\
\;  TYPE "? CR " FOR HELP.\
```

*Backslash ("\") is the NETWRK prompt and escape character. Type it anytime to return to the NETWRK command interpreter.*

7. To get help from the NETWRK subsystem, type:

```
\? [CR]
TYPE "HELP CR" FOR ADDITIONAL INSTRUCTIONS.
IMPORTANT COMMANDS:
-------------------------------------------
?  ??  HELP  SURVEY  REAL.TIME.SURVEY  STATUS  CONNECTION. TO  DISCONNECT
QUIT  LOGOUT  RESET  SOCKET.MAP  HISTORY.OF  SUMMARY.OF.SURVEYS  ACTIVE.HOSTS
BEST.SURVEY  CURRENT  LONGTERM  TIME  COMMANDS  ........
```

DMCG has a SURVEY subsystem which performs a periodic survey of ARPANET serving HOSTs to develop statistics on their availability. A SURVEY is taken automatically every 15 minutes and the results kept in a file for later study. The NETWRK subsystem allows on-line access to the SURVEY data base on serving HOST availability. Note that the SURVEY program runs every 15 minutes, 24 hours a day, when the DMCG ITS time-sharing system is in operation.

8. To print the results of the last survey taken:

```
\sur [SP] VEY

SURVEY TAKEN AT 18:03:39 on 09/19/72
--HOST--  -#-  ----STATUS----------
UCLA-NMC  001  LOGGER AVAILABLE.
SRI-ARC   002  LOGGER AVAILABLE.
UCSB-75   003  LOGGER AVAILABLE.
UTAH-10   004  LOGGER AVAILABLE.
............
```

*The list continues. The HOST numbers are given in octal (sorry). The various conditions reported describe the state of the HOST hardware, ARPANET Network Control Program, and TELNET server (LOGGER), all of which must be functioning to some minimum extent for "Logger available" to be reported.*

9. To get a brief summary of the SURVEYs for about the last 24 hours, type:

```
\sum [SP] MARY.OF.SURVEYS
00072 SURVEYS FROM 00:16:35 on 09/19/72 to 18:03:19 on 09/19/72
--HOST--  -#-  -%-UP-  -RESP-
UCLA-NMC  001  097%    00.80
SRI-ARC   002  068%    01.23
UCSB-75   003  059%    00.63
........
```

*The per cent gives the fraction of surveys which resulted in a "Logger available" condition for the specified serving HOST. The "resp" gives the mean number of seconds (for successful connection attempts) from the first probe to the first response for the specified HOST.*

10. To get a summary of SURVEY statistics for the "long term", type:

\lon [SP] GTERM sum [SP] MARY.OF.SURVEYS

```
10848 SURVEYS FROM 19:48:24 on 04/27/72 to 00:01:34 on 09/19/72
--HOST--  -#-  -%-UP-  -RESP-
UCLA-NMC  001  071%    01.00
SRI-ARC   002  058%    02.53
UCSB-75   003  053%    00.80
UTAH-10   004  062%    02.33
.......
```

*The list continues giving summary data for over 10,000 SURVEYs for about the last 6 months. Please note that this data does not account for scheduled down time, nor does it account for time that the specified HOST was up but not available to the ARPANET.*

11. To get a history for a specific serving HOST, type:

\lon [SP] GTERM his [SP] TORY.OF nic [CR]

*You can ask for a history for any of the serving HOSTs by name (see HELP) or by number (use octal or decimal preceded by period). Note that the "response time" measure given relates to the mean time (on successful connections) from the first probe to the first response, only.*

```
FIRST SURVEY AT 19:48:24 on 04/27/72
UNDETERMINED 00000 TIMES   (000%)
HOST DISCONNECTED 03474 TIMES   (032%)
NCP NOT RESPONDING 00000 TIMES   (000%)
LOGGER NOT RESPONDING 01016 TIMES   (009%)
LOGGER REJECTING 00000 TIMES   (000%)
LOGGER AVAILABLE 06358 TIMES   (058%)
AVERAGE RESPONSE TIME = 02.53
LAST SURVEY AT 00:01:34 ON 09/19/72 \
```

12. To exit from the NETWRK subsystem, type:

\ quit [CR]

```
KILL
```

13. To logout of the DMCG system:

; logout [CR]

```
ITS 761 CONSOLE 31 FREE
```

14. To disconnect from DMCG ITS, type:

@c [LF]

```
T R CLOSED
```

SPEAKEASY === HOST #65

The statement immediately below is an example in SPEAKEASY, an interactive language for researchers that is now available to ARPA Network users on the UCLA 360/91.

*X = MATRIX (2,2: 1,3,4,2); 1/X; EIGENVALUES(X)*

The above is all the information necessary to calculate and display the inverse and the eigenvalues of the indicated 2 x 2 matrix.

1. To connect to the 360/91 at UCLA, type:

```
@r [LF]                 Reset TIP, terminal-dependent setup here.
@t [SP] o [SP] L [LF]   "Transmit on linefeed", TSO is line-at-a-time.
@i [SP] L [LF]          "Insert linefeed", send a linefeed with each carriage-return.
@L [SP] 65 [LF]         Connect to UCLA CCN (Campus Computing Network).
UCLA CCN 360/91 SERVER TELNET.
VERSION X.X DD DDD 1972
ENTER COMMAND OR 'HELP':
```

2. To connect to TSO, type:

```
tso [CR]
WELCOME TO UCLA CCN TSO
IKJNJJJJA ENTER LOGON -
```

3. To LOGON to TSO, type:

```
logon [SP] icX [CR]     TSO echoes LOGON line. X is digit 1-9.
LOGON icc
ENTER PASSWORD
! iccc [CR]
IKJNNNNI UUU LOGON IN PROCESS AT T:TT:TT ON DD DD, 1972
.................
READY
!
```

4. To start SPEAKEZ, type:

```
! speakez [CR]          A short pause here for SPEAKEZ startup.
TSO-SPEAKEASY-3D T:TT AM D/DD/72
:←!                     The prompt character ":←!" indicates SPEAKEASY is now active
                        and awaiting input from the user.
```

Data is entered into the system line by line and may be entered in either upper or lower case. Each line is executed after a carriage return is transmitted.

5. To use SPEAKEZ, try the following interactions:

Please wait for the prompt character ":←!" to appear before entering successive lines of input. Output will be generated for many of the input lines indicated.

```
:←! x=5 [CR]
:←! x*9 [CR]
X*9 = 45

:←! x*x; x**3 [CR]      Multiple statements are separated by semicolons.
```

```
X*X = 25
X**3 = 125
```

Elements of an array can be entered on a single line, with commas separating the entries:

```
:←! y=1,-9.2,sqrt(2), ... [CR]          Enter some numbers of your own.
:←! y [CR]                              Print out the value of Y, a vector.
:←! y*9; y/x;y*y [CR]                   Various operations on a vector.
.........
:←! domain(complex) [CR]                Enable complex arithmetic.
:←! sqrt(y) [CR]
SQRT(Y)  (A 5 COMPONENT ARRAY)
   1      1.0332I   1.1892     .....    .....
```

6. To get information on any one of the over 200 SPEAKEZ words, use the HELP command as in the following example:

```
:←! help [SP] matrix [CR]  Helpful description of matrix manipulation is typed.
.........
:←! m=matrix (3,3: ......) [CR]        Enter up to 9 numbers separated by commas.
:←! m [CR]                              Shortened form of "print(m)".
:←! 1/m [CR]                            Shortened form of "inverse (m)".
:←! a=m*transpose(m);a [CR]
.........
:←! eigenvals(a) [CR]
```

7. To experiment, try some of the following:

   trace(a); det(a); max(a); min(a); average(a); sum(a); sumsq(a).

Information on any of these "words" can be obtained by typing:

```
:←! help [SP] "word" [CR]          @s [SP] S [LF] stops printout and gets
                                   system's attention.
```

8. To terminate SPEAKEZ, type:

```
:←! quit [CR]
CORE USED 1 K now, 1 K PEAK, ALLOCATED 4 K
.........
READY
!
```

9. To leave TSO, type:

```
! logoff [CR]
```

10. To sever your ARPANET connection, type:

```
! @c [LF]
T R CLOSED
```

About SPEAKEASY

Attempting to converse with a modern large scale computer can be a quite difficult and frustrating experience. As a result, researchers desiring to utilize the enviable power and resources of a computer are oftentimes understandably turned off by rigid machine restrictions.

SPEAKEASY is a language for people, not machines. The ability to converse with a computer in a notation similar to that of normal mathematics, rather than some foreign language, is SPEAKEASY's forte. A powerful vocabulary of commonly used operations is at the fingertips of the user, and aids to its usage are a very significant built-in capability of the system.

SPEAKEASY is easily learned, easily used, and its ability to relieve the user of trivial tasks associated with writing conventional computer programs, makes man-machine interaction the enjoyable experience it should be.

The system was developed at Argonne National Laboratory under the direction of Dr. Stanley Cohen of the Physics Division. It has been successfully implemented at several installations around the country under OS/360 for operation on the IBM 360/370 series computers. Now a valuable resource on the ARPA Network, SPEAKEASY is accessible via the 360/91 at UCLA.

To obtain more information on SPEAKEASY:

Questions regarding SPEAKEASY on the ARPA Network should be directed to:

Lawrence McDaniel (NIC Ident=LM)
314 Center for Advanced Computation
University of Illinois
Urbana, Illinois 61801
(217) 333-8497

or Ernest H. Forman
MITRE Corporation
Westgate Research Park
McLean, Virginia 22101
(703) 893-3500 X-2523

To request an account at UCLA, contact:

Mark Cirlin
Campus Computing Network
UCLA
Los Angeles, California 90074
(213) 825-7426

## BBN TENEX === HOST #69

Tenex is a PDP-10 time-sharing system developed at Bolt Beranek and Newman of Cambridge, Massachusetts (BBN) and now used at several ARPANET sites.

The attention character in Tenex is control-c (denoted "↑C"). In Tenex EXEC and most subsystems, control-a ("↑A") deletes the last character typed and control-x ("↑X") deletes the entire current line. In the EXEC and most subsystems, commands and filenames can be specified with the minimum number of characters assuring uniqueness, followed by ESCAPE (i.e., ALTMODE, denoted [ESC]; SPACE, denoted [SP]; or carriage return, denoted [CR]). Using ESCAPE will cause Tenex to complete a partially typed but uniquely specified command or filename.

In EXEC (Tenex command level), Tenex does not distinguish between upper and lower case alphabetics. Lower case input is echoed in upper case in EXEC and most subsystems. Over the ARPANET, Tenex does its own echoing, a character at a time.

1. To connect the TIP to BBN Tenex, type:

   @r [LF] *Reset the TIP, terminal-dependent setup here.*

   @L [SP] 69 [LF] *Cause TIP LOGGER to connect to BBN, HOST #69.*

   BBN-TENEX 1.29.6, SYSTEM-A EXEC 1.43

2. To login to Tenex, type:

   @ login [SP] iccc [CR] *At sign ("@") is Tenex's prompt character. Login with user name "iccc".*

   (PASSWORD) iccc [CR] *Use the password "iccc"; it will not print.*

   (ACCOUNT #) 11514 [CR] *Use account number 11514.*

   JOB 1 ON TTY103 2-SEP-72 16:39

3. To print a list of EXEC commands:

   @ ?

   COMMANDS ARE:

   ACCOUNT

   APPEND

   ..... *List continues. Can be stopped with control-c ("↑C").*

4. To send a message to another user:

   @ sndmsg [CR]

   TYPE LIST OF USERS: iccc [CR] *SNDMSG is self explanatory.*

   TYPE MESSAGE. EDIT WITH CONTROL-A, Q, R, X END WITH CONTROL-Z. INSERT A FILE WITH CONTROL-B.

   hello, this is a test. [CR]

   ↑Z

   End message with control-Z

5. To print a message:

   @ typ [ESC] E (FILE) message.tx [ESC] T;1 [CR] *ESCAPE (i.e., altmode, denoted [ESC]) causes completion of TYPE command and filename MESSAGE.TXT;1.*

```
;  <ICCC>MESSAGE.TXT;1   SAT 2-SEP-72 4:42PM          PAGE 1
2-SEP-72 1642 ICCC
HELLO, THIS IS A TEST MESSAGE.
```

6. To list status of users and jobs:

```
@ sys [CR]
UP 87:19:48:  3 JOBS
LOAD AV   0.34    0.45   1.00
JOB  TTY  USER       SUBSYS
1    103  ICCC       EXEC
9    101  TEITELMAN  LISP
12   7    TOMLINSON  (PRIV)
.....
```

*List continues. Can be stopped by typing control-c ("↑c").*

7. To find about an individual user:

```
@ where [SP] tomlinson [CR]
TTY7, JOB 12
```

*The specified user's terminal number and job number are returned if he is logged in to the time-sharing system.*

8. To link to the user at TTY7:

```
@ link [SP] 7 [CR]
LINK FROM ICCC, TTY 103
@ ;hello ray. [CR]
@;HELLO, WHO IS IT?
@ ;this is abhay bhushan from mit, can you help me? [CR]
@;OF COURSE, HOW ARE YOU ABHAY?
@ .....
@ ;nice talking to you. bye. [CR]
@ break [CR]
```

*Messages over "link" should be preceded by semi-colon.*

*Questions and comments from TTY7.*

*Conversation continues.*

*Break the "link", please.*

9. To list files in a directory:

```
@ dir [ESC] ECTORY <hacks> [CR]
   <HACKS>
CHESS.SAV;2
CORDERMAN.LIFE;1
.....
```

*Get a listing of the <HACKS> directory which contains programs to play various games.*

10. To list the <ICCC> directory:

```
@ dir [CR]
  <ICCC>
1  .F4 ;14,13
.....
```

11. To get the text editor "TECO", type:

```
@ teco [CR]
```
*The TECO prompt character is asterisk ("*") and TECO commands are terminated with ESCAPE (i.e., altmode, denoted ESC).*

```
* i [CR]
```
*"Insert" in a text buffer all type-in up to the next ESCAPE (ESC).*

```
 [TAB] type 100 [CR]
```
*Note that TAB is typed control-i ("↑I") or "HT".*

```
100 [TAB] format (' [SP] hello [SP] iccc.') [CR]
 [TAB] end [CR]
 [ESC]
```
*You type ESC or ALTMODE which is echoed as '$'.*

```
* ht [ESC]
      TYPE 100
100   FORMAT (' HELLO ICCC.')
      END
```
*"Type" the "whole" text buffer.*

*You may edit any typing errors detected. See notes on using TECO at the end of this scenario.*

```
* ;U [ESC]
OUTPUT FILE:  akb.f4 [CR] (NEW FILE) [CR]
```
*"Unload" text buffer onto a disk file.*

*You name the file and type CR twice, once to confirm.*

```
* ;h [ESC]
```
*"Halt" TECO and return to EXEC.*

```
@ typ [ESC] E  (FILE)  akb.f4 [ESC] ;1 [CR]
```
*Type the FORTRAN program just entered using TECO*

```
;  <ICCC>AKB.F4;1   SAT 2-SEP-72 5:36PM          PAGE 1
      TYPE 100
100   FORMAT (' HELLO ICCC.')
      END
@ f40 [CR]
```
*Start the FORTRAN compiler.*

```
* akb+akb [CR]
```
*Give the source file name AKB, AKB.F4 assumed. The compiled 'object' file will be AKB.REL.*

```
MAIN.  ERRORS DETECTED:  0
8K CORE USED
* ↑C
```
*Compilation is performed and no errors are detected. You are returned to the EXEC after typing control-c.*

```
@ loader [CR]
```
*Relocatable output from compiler must now be loaded for execution.*

```
* akb [CR]
```
*You type program name to the loader and it gets its input from AKB.REL for loading.*

```
* [ESC]
```
*Typing ESCAPE causes loader to do a library search for subroutines used by your program and to exit to EXEC.*

```
LOADER 2K CORE
5+4K MAX 586 WORDS FREE
EXIT.
↑C
@ save [ESC]  (CORE FROM)  0 [SP] 777777 [SP] akb.sav;1 [ESC]  (NEW FILE) [CR]
```
*Save the core image generated by the loader in file AKB.SAV for future execution.*

```
@ run [SP] akb [CR]
```
*Run the sample FORTRAN program.*

```
HELLO ICCC.
CPU TIME:  0.05 ELAPSED TIME:  1.00
NO EXECUTION ERRORS DETECTED
EXIT
↑C
```
*The program performs as expected.*

12. To delete all files we created to clean up ICCC directory:

```
@ delete akb.* [CR]          Delete all files whose first name is AKB.
```

13. To access to the ARPANET:

```
@ telnet [CR]                Start the TELNET (TELecommunications NETwork) subsystem.
USER TELNET 29 MAR 72.  TYPE HELP CR FOR HELP.
                             "#" is the TELNET prompt character.
```

14. To see status of ARPANET:

```
# netstatus [CR]             By typing control-z ("↑Z") you can force a return to the
                             TELNET command interpreter.
THE FOLLOWING HOSTS ARE UP:
UCLA-NMC
UCLA-CCN
UCSB-MOD75
UTAH-10
BBN-TENEX
MIT-MULTICS
MIT-DMCG
MIT-MATHLAB
.........                    List continues.
ACTIVE CONNECTIONS:          A list of ARPANET connections to Tenex is provided with
                             their conditions.  The table needs some explaining not
                             included here.
#  ?                         Question mark will list TELNET commands and key words.
                             To stop the long list, type control-z ("↑Z") and return to
                             TELNET command interpreter.
CONNECTION.TO
DISCONNECT
.........                    List continues.
# quit [CR]                  "Quit" causes the TELNET subsystem to return you to the
                             Tenex EXEC.
```

15. To see the date and time:

```
@ day [CR]
SATURDAY SEPTEMBER 2, 1972 17:00:20
```

16. To log out of Tenex:

```
@ logout [CR]
KILLED JOB 1 USER ICCC, ACCT 11514, TTY 103 AT 9/2/72 15:01
  USED 0:0:9 IN 0:22:10
```

17. To disconnect TIP from BBN Tenex, type:

```
@c [LF]
T R CLOSED                   "Transmit" and "Receive" closed.
```

---

NOTES ON USING TECO:

1. Move current text pointer by the following commands:
   a) bj [ESC] — *moves pointer to beginning of buffer.*
   b) s <STRING> [ESC] — *searches for the specified <STRING> and moves pointer to just after it.*
   c) nl [ESC] — *moves pointer to nth line from current line, where n is a positive or negative integer.*
2. Delete the text by the following commands:
   a) nk [ESC] — *will kill n lines starting from the current position.*
   b) nd [ESC] — *will delete n characters from the current position*
3. Insert text by the I command, i.e., i <STRING> [ESC] .
4. Example, to change ICCC to THERE in above program, the following would work:

   bj [ESC]
   sICCC [ESC]
   -4d [ESC]
   iTHERE [ESC]
   ht [ESC]

MIT H645 MULTICS === HOST #6

Multics interacts line-at-a-time and assumes local echoing at the TIP. Multics distinguishes between upper and lower case alphabetics. To gain an understanding of the concepts behind this general purpose time-sharing system, it is recommended that you attend one of the scheduled demonstrations. See steps 4 and 9 for special characters to delete character or signal system.

1. To connect to Multics, type:

```
@ r [LF]                          Reset the TIP, terminal-dependent setup here.
@ t [SP] o [SP] L [LF]            Line-at-a-time, "transmit on linefeed".
@ i [SP] L [LF]                   Insert linefeed after carriage-return.
@ L [SP] 6 [LF]                   TIP command to connect to Multics.
Multics 17.6b; MIT, Cambridge, Mass.
Load = 28.5 out of 50.0 units; users = 30
```

2. To login to Multics, type:

```
enter [SP] NAME [SP] CNet [CR]    Note upper case "CN" and lower case "et" in CNet.
                                  Please use your last name for "Name".
.....                             Message from the system.
r 1148   6.437   2+588            The ready message is printed at the end of processing
                                  each command. The numbers represent time of day,
                                  cpu time for last command, and pre-paging + page
                                  faults, respectively.
```

3. To get specific on-line help, type:

```
'hello from iccc [CR]             Statements prefixed with an apostrophe will be sent
                                  to a network consultant for on-line help or to a file
                                  for later study by a consultant.
```

4. To print help file for pl/1, type:

```
help [SP] pl1 [CR]                Number-sign ("#") deletes the previous character in an
                                  input line and at sign ("@") (typed "@@" at a TIP)
                                  deletes the entire line.
```

5. Help may be used with most commands, e.g. type:

```
help [SP] who [CR]
```

6. To see who is on the system:

```
who [CR]
multics 17.6b, load 30.5/0.0; 32 users
absentee users 2/2
roach.sysmaint*
network daemon.CNet
IO.sysdaemon
backup.sysdaemon
carey.mpm
.
.
.                                 List continues.
r 1150   4.039   15+42
```

7. To print the working directory:

```
print_wdir [CR]                    Can be abbreviated pwd [CR].
>udd>CNet>anonymous                Pathname associated with your process's working
r 1151   .757   16+38              directory.
```

8. To list segments in the working directory:

```
list [CR]
Segments = 66, Records = 109.
r wa   2   testl
r wa   0   mailbox
.
.
.                                  List continues.
r 1151   9.438   52+118
```

9. To get attention from Multics:

```
@ s [SP] s [LF]                    TIP command to send synch.
                                   Equivalent to TTY BREAK or 2741 ATTN.
QUIT
r 1152   .648   14+14
```

10. To send mail to specified user:  *In this case, Padlipsky:*

```
mail [SP] * [SP] Padlipsky [SP] CompNet [CR]
Input
Hello Mike, this is a sample mail for the Multics scenario. [CR]
. [CR]                             A line containing only a single period
                                   terminates and sends mail.
r 1156   3.673   113+152
```

11. To see if anyone has sent you mail:

```
mail [CR]                          Reads mail sent to "anonymous CNet".
No mail now.
r 1156   1.184   24+83
```

12. To logout:

```
logout [CR]
Name CNet logged out 09/12/72   1203.3 edt Tue
CPU usage 46 sec
hangup
T CLOSED R CLOSED                  You are automatically disconnected from Multics.
```

## SRI-ARC (NIC) === HOST #2

The SRI-ARC Online System (NLS) is a powerful system being designed to provide aids to a wide variety of general intellectual tasks; for a more complete demonstration of its capabilities see the schedule of demonstrations posted in the Ballroom. SRI-ARC provides both online and offline services to the Network as the ARPA Network Information Center (NIC) with a DEC PDP-10 computer running the BBN Tenex timesharing system.

The following characters are of importance:

CONTROL CHARACTERS

The up-arrow character ↑ when followed immediately by another character means input a control character. A control character is achieved by pressing the Control, CTRL, or Shift II (depending on your terminal) key IN CONJUNCTION WITH the character specified. See the following control characters.

↑c -

TO RETURN CONTROL TO THE TENEX EXECUTIVE SYSTEM TYPE ↑C.

↑a -

IF YOU MAKE A TYPING ERROR AT ANY POINT, type ↑a to backspace one character. One character is deleted each time this code is entered.

↑t -

IF YOU WANT TO CHECK TO SEE IF THE SYSTEM IS STILL THERE, type ↑t. The system will respond with "RUNNING AT ---" if it is executing your command. It will respond with "I/O WAIT AT ---" if it is expecting input from you. No response means the system is down.

↑o -

TO STOP NLS PRINTING type ↑o and you will be returned to the command level.

DEL -

DEL is the DEL, Delete, or Rubout key on your terminal.

IF YOU AREN'T SURE OF WHAT YOU ARE DOING DURING A NLS COMMAND SPECIFICATION, type DEL and NLS will return to the command level. This is the Command Delete character (CD).

CR -

is the Carriage Return or Return key on your terminal. CR is used in NLS as a field delimeter and as Command Accept character (CA).

SP -

SP stands for space. Spacing on the paper is for readability only, whenever you are to explicitly type a space it says SP enclosed in a box.

? -

TO OBTAIN MORE INFORMATION ABOUT A PARTICULAR NLS COMMAND type the character "?" at any point during command specification.

ACCESSING THE SRI-ARC SYSTEM

1. To connect to SRI-ARC through the TIP type the following sequence: (NOTE: You must type the character @ before giving any command to the TIP. This is not to be confused with the TENEX prompt character "@" which is printed by the system when it is awaiting a command from the user.)

| | |
|---|---|
| @r [LF] | *Reset the TIP. Terminal-dependent setup here. Consult the parameter card attached to your terminal for appropriate TIP commands, if any.* |
| @L [SP] 2 [LF] | *Cause the TIP "Logger" to connect you to SRI-ARC, HOST #2.* |

LOGGER

R OPEN T OPEN — *Reply by the TIP that it has established a connection.*

ARC-TENEX 1.29.00 DATE ARC EXEC 1.38.00

..... — *Various SRI-ARC messages will be typed here. When the system is ready to accept commands it will type its prompt character "@".*

2. Setting up the SRI-ARC system for your terminal. If you are at a full duplex terminal type:

   @full [CR] — *On initial entry, Network users see only the characters they type. Using this command enables you to see characters echoed by the SRI-ARC system.*

3. To log into the Tenex system at SRI-ARC:

   @log [CR]
   (USER) iccc [CR]
   (PASSWORD) iccc [CR]
   (ACCOUNT #) 3 [CR]
   JOB # ON TTY# DATE TIME

SCENARIO FOR THE ON-LINE RESOURCE INFORMATION RETRIEVAL SYSTEM (Q1)

This scenario demonstrates the application of a simple information retrieval system to a developing data-base of network facilities. This on-line service provides users at any ARPANET site with three types of data:

i) Indices of computers, terminals, and programs on the ARPANET.

ii) Site-oriented data giving detailed information about that installation's software, hardware, and service configuration, as well as staff names and phone numbers, and

iii) "Help" information on the use of the data-base.

4. To use the NIC Resource Query System:

   @nic [ESC] (Resource Query) [CR] — *At this point the retrieval system is initialized. The Q1 prompt character is a hyphen (-).*

   TYPE ? IF YOU NEED HELP AT ANY POINT.

5. To list available options:

   -? — *At any point (even in the middle of a command) the user can type a question mark to obtain information about available options. For example, if you type,*

   - show ? — *Instructions for use of the Resource Notebook Data Base will be typed.*

   *NOTE: WHILE ENTERING A COMMAND THE CHARACTER ↑a CAN BE USED TO ERASE THE LAST CHARACTER TYPED AND THE CHARACTER DEL WILL RETURN YOU TO THE PROMPT SYMBOL -.*

   When you have seen enough of the list, type a control-o (↑o) to interrupt printing.

6. Site information is broken into main categories. To browse through this information the user will type, for example:

```
-show sri-arc [CR]
(SRI-ARC)              Stanford Research Institute
                       Augmentation Research Center (ARC)
                       Network Information Center (NIC)

Choose one by typing, for example:  show personnel  CR
(FUNCTION)
(ADDRESS)
(PERSONNEL)
(HARDWARE)
(SOFTWARE)
(INTERESTS)
(DOCUMENTATION)
```

Given the list of topics about that site you can pick one by typing for example:

```
-show hardware [CR]
```

*Information about that topic (possibly in the form of a list of secondary options) will be printed. One can directly access information about a specific topic and site by typing the site ident followed by a colon and the topic as shown in steps 9 and 10 below.*

7. To display a table of all the computers available on the Network:

```
-show computers [CR]
```

8. To display a table of programs available online:

```
-show programs [CR]
```

9. To display the interests of a particular site:

```
-show mit-dmcg:interests [CR]
(INTERESTS)
```

*Material describing MIT-DMCG's interests will be typed.*

Type ↑O to stop printing at any point.

10. To display the personnel at a particular site:

```
-show sri-arc:personnel [CR]
(PERSONNEL)
  STATION AGENT...
```

11. To return to the EXEC:

```
-quit [CR]
```

SCENARIO FOR NIC DOCUMENT LOCATOR AND BROWSING SYSTEM

This scenario demonstrates use of NLS to access and browse in selected documents online. Locator is normally used by people with some knowledge and experience in using NLS.

12. To enter the online (NLS) system at SRI-ARC:

    @nls [CR]

    *When NLS is ready for you to type it will print its prompt character, "*".*

13. To access LOCATOR:

    *load file <nic>locator [CR]

14. To list documents that you can reach with Locator:

    *print branch .2 [CR]

    xbm [CR]

    *xbm are codes which tell the system to print only the parts of the file you need to see now.*

    *The numbers and letters preceding the name of each document are NLS STATEMENT NUMBERS.*

15. To see the table of contents for a specific document, use the print branch command and indicate the statement number of the document you want to see preceded by a period:

    *print branch .STATEMENTNUMBER [CR]

    xeb [CR]

16. You can use each item in the table of contents list to reach a file containing that part of the document. To load and print a particular file, type:

    *print branch .STATEMENTNUMBER [SP] ↑ [CR]

    [CR]

    *The character ↑ is to be literally input in this step and does not signify a control character.*

    *Some terminals have a circumflex (^) instead of up arrow (↑). They do the same thing.*

    The new file will print out either short text, or instructions for how to proceed.

    At the end of the printout, the system will supply the name of the new file in a special format, e.g. <nic>LOC7440.nls;8

17. To return to LOCATOR:

    *[SP] & [CR]

    *After execution of this command you will be back in LOCATOR where you were before going to the selected document. You can now continue to browse in other documents by returning to step 14 above.*

18. To leave NLS and return to the TENEX EXECUTIVE for the next scenario type:

    *quit [CR]

A SCENARIO DEMONSTRATING THE SHARING OF A FILE STORAGE RESOURCE

This scenario demonstrates the use of extra file storage capacity at the University of California, Santa Barbara to be used by SRI-ARC for archival purposes using a system called Simple Minded File System (SMFS).

19. To view the file to be sent to UCSB:

    @copy [SP] <system>sample [ESC] .TXT;1 tty: [CR] [OK] [CR]

    *One paragraph of text -- the contents of the file -- follows.*

20. To enter SMFS at SRI-ARC:

    @smfs [CR]

    UCSB Archival System (ver 1.0 6-SEP-72)

    *Message from SMFS: The SMFS herald character is "#".*

21. To copy the file to UCSB:

    @cOPY

    (TO/FROM UCSB) tO

    (FILE) <system>sample [ESC] .TXT;1 [CR]

    (CREATE/REPLACE) cREATE

    *When the transfer is complete, SMFS will respond with its prompt character, at which point a copy of the file will exist at UCSB.*
    *If the message HOST NOT OPERATIONAL should appear then go to step 29.*

22. To verify that a copy exists at UCSB:

    #lOCATE (FILE) [ESC] <SYSTEM>SAMPLE.TXT;1 [CR]

    Archived at UCSB

23. To rename that copy:

    #rENAME (FILE) [ESC] <SYSTEM>SAMPLE.TXT;1 [CR]

    (NEW FILE) YOURLASTNAME.txt;1 [CR]

24. To verify that the file has been renamed at UCSB:

    #lOCATE (FILE) <system>sample [ESC] .TXT;1 [CR]

    Not archived at UCSB

    #lOCATE (FILE) YOURLASTNAME.txt;1 [CR]

    Archived at UCSB

25. To return the renamed file to SRI-ARC:

    #mOVE

    (TO/FROM UCSB) fROM

    (FILE) [ESC] YOURLASTNAME.TXT;1 [CR] *Two copies of the file now exist at SRI-ARC: the original and the renamed version. The copy at UCSB has been deleted.*

    #lOCATE (FILE) [ESC] YOURLASTNAME.TXT;1 [CR]

    Not archived at UCSB

26. To leave SMFS:

    #qUIT [CR]
    @

    *This is the system's prompt character.*

27. To verify that a second copy of the file exists at SRI-ARC:

    @copy [SP] YOURLASTNAME [ESC] .TXT;1 (TO) tty: [CR] [OK] [CR]

    *The text of the file follows again.*

28. To delete the second copy you created at SRI-ARC:

    @del [SP] YOURLASTNAME [ESC] .TXT;1 [CR]

29. To leave the SRI-ARC system type:

    @logout [CR]
    TERMINATED JOB #, USER ICCC, ACCT 3, TTY 52, AT 8/25/72 1453
    USED 0:0:19 in 0:12:0

30. To disconnect from host SRI-ARC:

    @c [LF]
    T R CLOSED

## HARVARD PDP-10 === HOST #9

The Harvard system interacts character-at-a-time and initially assumes local echoing at the TIP. The attention getting character is ETX, i.e., control-C. The prompt character is period. DEL or Rubout deletes the previous character. ↑o stops printout.

1. To reset TIP parameters:

   @ r [LF]  *Terminal-dependent parameters set here.*

   @ i [SP] L [LF]  *TIP to insert LF after CR.*

2. To connect to Harvard:

   @ L [SP] 9 [LF]

   LOGGER

   T R OPEN

   Harvard 5S0401-31X 16:54:22

   Please LOGIN or ATTACH, or type HELP for help.

3. To obtain help:

   .help [CR]  *The HELP command prints helpful documentation for various commands and programs. 'HELP *' prints a summary of all help texts available. 'HELP NAME' prints the documentation for the NAME command or program.*

   .KJOB

4. To see what help exists:

   .help [SP] * [CR]

   attach  echo  help  impcom  kjob  login  logout

   .KJOB

5. To log into Harvard:

   . login [SP] 74,365 [CR]

   Job 10 Harvard 5s0401-31x TTY30

   password:

   iccc [CR]  *Mask is typed for password.*

   1657  DATE  DAY

6. To get system status information:

   .systat [CR]

   status of harvard 5S0401-31X at TIME on DATE

   uptime 28:09:04, 86%null time = 85%idle +1%lost

   9 jobs in use out of 12.  9 logged in, 3 detached

   job  who  line#  what  size(K)  state  run time

   *The long printout continues.*

7. To use PPL: *See manual for complete instructions.*

```
.r [SP] ppl [CR]
PPL H.47(133) 17-SEP-72
4+4 [CR]
8                    PPL evaluates the expression typed.
20*30 [CR]
600
40+40*50 [CR]
2040                 Evaluation is from right to left.
s←"this is a string" [CR]
s[4] [CR]
S                    Print 4th element of S
s[3]←'a [CR]
s[4]←'t [CR]
s [CR]
THAT IS A STRING
$complex=[rp:real,ip:real] $ [CR]   Creates constructor, selector, and
                                    predicate, below.
z←complex(3.2,5.6) [CR]             Constructor used to make data of specified type.
z [CR]
[RP:3.2,IP:5.6]
rp(z) [CR]           Get the "real part" of z. Selectors used to get at parts of
                     data using definition.
3.2
ip(z) [CR]           Get the "imaginary part" of z.
5.6
z==real [CR]
FALSE                The predicate "z is real" is false.
z==complex [CR]
TRUE                 The predicate "z is complex" is true.
```

8. To exit PPL, type control-C, i.e., ETX.

   *On a teletype, holding down <CTRL> key, strike C key.*

9. To leave the Harvard system, kill your job:

```
.kj [CR]
CONFIRM: k [CR]   K will delete unprotected files.
job #, user [74,365] logged off tty #, 1717    DATE
runtime # min, # sec
```

   *If the computer asks you to CONFIRM at this point, just type a [CR] .*

10. To disconnect:

```
.@c [LF]
```

## SAIL AP Hotline === HOST #11

The AP Hotline is a direct Associated Press news line carrying national and international news. The AP Hotline has been interfaced to the SAIL system at the Stanford Artificial Intelligence Laboratory. Any terminal on the ARPA Network can be turned into an AP news line by running program "HOT" at SAIL.

In addition, the APE system at SAIL processes the AP Hotline continuously, collecting its stories into an on-line data base of news information. This data base can be accessed via a keyword system by running the program APE at SAIL.

SAIL prefers to do its own echoing, a character at a time. Its attention getting character is control-c ("↑C"). When a program is expecting input, typing one ↑C will cause a return to the command level. Typing two ↑C's will cause an unconditional return to the command level, even during program output. To delete the previously typed character on input, type DEL or RUBOUT. To delete an entire input line, type control-u ("↑U"). The executive command interpreter uses period (".") as its prompt character. It has been observed that the system will type "TIMEOUT" on a few spurious occasions; typing carriage return ([CR]) has been observed to bring good results.

1. To set up the TIP to talk to SAIL, type the following:

| | |
|---|---|
| @e [SP] r [LF] | *"Echo remote", SAIL prefers to echo, character by character.* |
| @i [SP] L [LF] | *"Insert linefeed" after every carriage return.* |
| @L [SP] 11 [LF] | *Cause TIP "Logger" to connect to SAIL, HOST #11.* |
| LOGGER | *TIP says you are being connected.* |
| T R OPEN | *TIP says you are connected, both "Transmit" and "Receive".* |

2. After you have been connected to SAIL by the TIP, you must log in. SAIL may type out a number of messages before you can login, so be patient. Typing ↑C's will suppress message output after a while. A typical interaction is shown below:

| | |
|---|---|
| SAIL PDP-10. | |
| ..... | *Messages of the day; they can be suppressed with ↑C's.* |
| . login [SP] ic,cc [CR] | *You may get a message back saying "are you sure?". If this happens, just type "yes CR ".* |
| JOB 27 STANFORD 6.09B 9-19-72 | |
| FRIDAY 22-SEP-720944 | |
| ↑C | |
| . | *You are now logged into SAIL; the "." is the executive system prompt character.* |

3. To run the Hotline program:

| | |
|---|---|
| . r [SP] hot [CR] | |
| ...ASSOCIATED PRESS NEWS... | |
| 026 | |
| EXPLORER SATELLITE 260 | |
| CAPE KENNEDY, FLA. (AP). A THREE-STAGE DELTA ROCKET IS POISED FOR AN ATTEMPT TONIGHT | |
| TO HURL... | *The AP Hotline will type news stories on your console as they come over the AP news line. If there are no news stories coming, your terminal will sit silently, waiting. At any time, you can type ↑C's to return to the executive command interpreter.* |
| ↑C | |
| ↑C | |
| . | *You have returned to the executive.* |

4. To run the APE program:

```
. r [SP] ape [CR]          Run APE, a program to give on-line access to a data base
                           of AP news information.
TYPE "?" AND RETURN AT ANY TIME FOR HELP.
KEYWORD EXPRESSION: ? [CR]
.................          A brief helpful message is typed.
KEYWORD EXPRESSION: nixon [CR]
12 NEWS ITEM(S) FOUND.  READ WHICH ONE(S)? 1 [CR]
                           APE will search its news data for stories which contain
                           the word "NIXON", it will tell you how many it has found,
                           and ask how many of them you would like to see.  And so on.
```

5. To logout of SAIL:

```
↑C                         Type control-c to return to the executive; it will prompt
                           with period (".").
. kjob [CR]                "Kill" your job to log out.
JOB 28, IC,CC LOGGED OFF TTY 122 10:50 22 SEP-72
.11 HOURS, CONSOLE TIME
.08 MINUTES, CPU TIME
.40 K AVERAGE CORE
KJOB
```

6. To close the SAIL connection:

```
@c [LF]
T R CLOSED
```

MIT-AI PDP-10 === HOST #134

The MIT Artificial Intelligence Laboratory runs its own ITS time-sharing system on a PDP-10. ITS prefers to do its own echoing, even cross-country.

1. To set TIP parameters and connect to MIT-AI:

| | |
|---|---|
| @r [LF] | *Reset the TIP, terminal-dependent setup here.* |
| @e [SP] r [LF] | *"Echo remote", TIP will not echo.* |
| @L [SP] 134 [LF] | |
| ↑z (or 198) | *Press Z while holding down the key marked "CTRL" or "CONTROL". This indicates to the system that you desire attention.* |
| ITS.761. DDT.460. | |
| 7. USERS | *At this point the system may type what is known as "the message of the day" which is usually of general interest to the everyday users of the system. It may be ignored, generally, without any loss of continuity.* |

2. To login to MIT-AI:

| | |
|---|---|
| :login [SP] iccXXX [CR] | *XXX should be your initials. This tells the system who you are and is necessary before you can proceed any further. Note, there is no prompt; you must type the ":" !* |

3. To run the program JOTTO:

:jotto [CR]

JOT COMPUTATION IS ONE-TO-ONE LETTER MATCHING

.....

WOULD YOU LIKE TO GO FIRST?

JOTTO is a word game played by two players: yourself and the program. Each player thinks up one five letter secret word. The object of the game is to guess your opponent's secret word by deducing which letters it contains. You do this by presenting your opponent with a five letter test word.

Your opponet then tells you how many of the letters in that word match the letters in his secret word. The message about one-to-one letter matching means that if your test word was "SISSY" and the program's secret word was "TEARS", the number of matching letters or "JOTS" would be one. The same is true if the test word was "TEARS" and the secret word was "SISSY". You and your opponent alternately give each other test words and number of "jots" until one of you guesses the other's secret word.

*Note: Carriage returns ([CR]) are not required to terminate your input. The system automatically responds to you after you type five characters to specify your word or one number to specify number of "jots".*

WOULD YOU LIKE TO GO FIRST?

yES *This particular interaction is for illustrative purposes only. JOTTO choses its own word differently each time.*

YOUR TEST WORD: stare

0 JOTS

MY TEST WORD: TEARS

1 JOT

```
YOUR TEST WORD:  mound
1 JOT
MY TEST WORD:  DEIGN
1 JOT
YOUR TEST WORD:  child
3 JOTS
MY TEST WORD:  SPILL
1 JOT
YOUR TEST WORD:  dilly
1 JOT
MY TEST WORD:  BONUS
0 JOTS
YOUR TEST WORD:  chimp
MOBY FOO, YOU WIN!  SHALL I KEEP GUESSING?
yES
MY TEST WORD:  LEECH
1 JOT
MY TEST WORD:  MADLY
1 JOT
MY TEST WORD:  ACIDS
2 JOTS
MY TEST WORD:  ALIKE
3 JOTS
IS YOUR WORD KHAKI?
yES
WOULD YOU LIKE TO GO FIRST?
```

*At this point the game starts over again. When you are tired of playing this game you may go on to the next one by typing the following:*

```
↑z
1451   .IOT 1,1
```

*This message indicates that at the time you typed ↑z, the program was waiting for a reply to its question.*

```
:kill [CR]
```

*This indicates to the system that you are through playing JOTTO and are ready to do something else.*

4. To run and use the program CHESS:

```
:chess [CR]
setd [SP] 1 [CR]
```

*This tells the CHESS program to look ahead only one move. The program is capable of playing a much better game by looking ahead several moves, however, this ties up a substantial portion of the system's resources and takes on the average five minutes per move. For the purposes of this demonstration, SETD 1 is just right.*

At this point you may choose either black or white. If you choose white then type:

```
pb [CR]
```

*This tells the program to play black.*

```
p-k4 [CR]
```

Moves are typed in standard chess notation. If you make a mistake while typing a move, simply type a RUBOUT and then start typing the move over again. The program will type out its moves as it makes them.

P/K2-K4  0.4 IN 1.0

*This is a typical first move for the program. This means that the program used 0.4 seconds of machine time and 1.0 seconds of real time to generate its move.*

At any time you may have the program type out the position by typing:

bd [CR]

The board will be typed out in a fashion similar to the following:

```
BR BN BB BQ BK BB BN BR
BP BP BP BP ** BP BP BP
-- ** -- ** -- ** -- **
** -- ** -- BP -- ** --
-- ** -- ** WP ** -- **
** -- ** -- ** -- ** --
WP WP WP WP -- WP WP WP
WR WN WB WQ WK WB WN WR
```

*-- indicates a white square and ** indicates a black square.*

The program also detects impossible or ambiguous moves and informs you of the fact. You may then proceed to type in a legal move.

At the end of the game, or if you get tired, you should type:

↑z

5. To log out of the system:

:logout [CR]

6. To disconnect from the AI system:

@c [LF]

A typical remote job entry capability is demonstrated by the following scenario of the Remote Job Service offered between Tenexes and the UCLA 360/91. A simple FORTRAN job created on a PDP-10 at BBN is submitted via a Remote Job Service (RJS) subsystem to an IBM-360/91 at UCLA. Various checks are made before the start and during the running of the job to determine its status. The output is then retrieved from UCLA and scanned at BBN.

1. To setup the TIP and connect to BBN, type:

```
@r [LF]                                  Reset the TIP, terminal-dependent setup done here.
@L [SP] 69 [LF]                          Connect to BBN, HOST #69.
LOGGER
R OPEN T OPEN
1       BBN-TENEX 1.29.6,  SYSTEM-A EXEC 1.43
```

2. To log into BBN, type:

```
@log [SP] iccc [SP]  iccc [SP] 11514 [CR]
                                         The password (i.e., the second "iccc") is
                                         not echoed.
JOB 20 ON TTY107  8-SEP-72  13:47
@ind [CR]                                Suppress formfeeds for the following listings.
```

3. To type out sample RJE FORTRAN submission, type:

```
@type [SP] fort.;1 [CR]
; <ICCC>FORT.;1  THU  31-AUG172 4:23PM     PAGE 1
//CPW502F JOB
//PASSWORD EBRAHIMI
//FORT.SYSIN DD *
      WRITE (6,40)
      FORMAT (35X,9H*HEADING*)
      DO 10 I=1,50
   10 WRITE (6,50) I
   50 FORMAT(1X,I2)
      STOP
      END                                This sample program causes the first fifty
                                         integers to be printed.
```

4. To start the BBN and UCLA 360/91 RJS subsystem, type:

```
@rjs [CR]
ENTER YOUR TERMID netanyt [CR]           Tenex requests an ID so it can login (for you)
                                         at the UCLA 360/91.
NRJ876I NETWORK REMOTE JOB SERVICE READY
RJS750I TERMINAL NETANYT HAS SIGNED ONTO RJS  The 360/91 joins the three-way conversa-
                                         tion (i.e., you, Tenex, and the 360/91) by
                                         announcing a successful connection and login.
RJS652I INFORMATION ALERT
....
RJS661I END OF SYSTEM ALERTS
```

TIPCOPY [CR]
name?
ICCC TIP [CR]
number? 32 file? JOBOUT

5. To get status of jobs submitted from this terminal, type:

```
/status [SP] jobs [CR]
```

*You type a command which is forwarded to the 360/91 requesting the status of any jobs which may have been submitted from your terminal previously.*

```
RJS783I TERMINAL STATUS CHANGED
RJS804I TERMINAL NETANYT HAS NO JOBS ACTIVE
```

6. To request list of all lines active into RJS, type:

```
/status [SP] lines [CR]
```

*Another status request is sent to the 360/91 to examine the current status of other ports into the 360/91 RJS system.*

```
RJS800I TERMINAL FORESTRY ACTIVE ON LINE2
RJS800I TERMINAL NETANYT ACTIVE ON LINE6
.....
RJS800I TERMINAL NETLL67 ACTIVE ON LINE10
```

7. To submit the sample job to RJS, type:

```
s END FROM fort.; [CR]
```

*You type this command to Tenex instructing it to cooperate with the 360/91 in the transfer of your job submission.*

```
11 SENT
RJS534I JOB CPW502F ACCEPTED BY RJS - 0000011 CARDS READ
```

*A total of eleven cards (the sample program above) were sent from Tenex to the 360/91 for submission as an RJS batch job.*

8. To check status of submitted job (as above), type:

```
/status [SP] jobs [CR]
RJS810I TERMINAL NETANYT HAS THE FOLLOWING JOBS IN RJS
RJS812I CPW502F  XEQ   000          Your job is in execution (XEQ).
/status [CR]
```

At some later time (possibly only a few tens of seconds) and possibly (but not typically) after a complete disconnection/reconnection with the 360/91, you will ask for status from the RJS system and find that your job has run and that the printable output is ready for retrieval.

```
RJS802I TERMINAL NETANYT HAS 1 XEQ JOB(S)
.........
.........
RJS48I PRINT OUTPUT FOR JOB CPW502F NOW AVAILABLE, PRTY=070,
/status [SP] jobs [CR]
RJS810I TERMINAL NETANYT HAS THE FOLLOWING JOBS IN RJS
RJS812I CPW502F  PRT   070
```

9. To retrieve RJE output, type:

```
prINT TO jobout [CR] NEW FILE [CR]
```

*You type this command to Tenex telling it to cooperate with the 360/91 to bring the output of your FORTRAN job back to Tenex for examination as file JOBOUT.*

```
r - PR=142
```

During transmission of your output from the 360/91 to Tenex, you can ask for a progress report if you grow impatient.

r - PR=258

...

289 PRINTED — *When the transmission of your output is complete, you are notified and the number of output lines delivered is indicated.*

qUITTING

Now that the output of your sample job has returned from the 360/91, you can QUIT using RJS and return to the Tenex system to look at the results. Because the output from RJS submissions is intended for line-printer processing, it is somewhat awkward to view results from an interactive terminal. You will now use an on-line editor (TECO) on Tenex to scan through the large output file for the desired results. Bringing such output to a line-printer connected to the TIP is possible and can be demonstrated.

10. To scan output with on-line editor, TECO, type:

@teco [CR] — *Instruct Tenex to start TECO.*

*:y [ESC] — *Request file input, ESCAPE (i.e., ALTMODE, denoted "ESC") terminates TECO commands.*

INPUT FILE: jobout [CR] CONFIRM [CR] — *Cause TECO to look into the RJS output file created above.*

16889 CHARS

2s*HEADING* [ESC] — *Search for the second occurrence of the string "*HEADING*" in the output file; this marks the beginning of the output desired.*

*20t [ESC] — *Type 20 lines of output (20 lines past the header).*

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 — *The first 20 lines of output are the first 20 integers, as expected from the submitted sample FORTRAN program.*

11. To logout and disconnect, type:

↑c — *Type control-c to return to the time-sharing system command level.*

@logout [CR]

KILLED JOB 20, USER ICCC, ACCT 1, TTY 107, AT 9//08//72 1400

USED 0:0:28 IN 0:13:41

@c [LF]

T R CLOSED

MATHLAB'S MACSYMA === HOST #198

MACSYMA (pronounced "maxima"), Project MAC's SYmbolic MAnipulation system, is a large computer program, written in LISP, devoted to the manipulation of algebraic expressions. MACSYMA runs under the ITS time-sharing system (originally developed at the MIT Artificial Intelligence Laboratory), on the Mathlab PDP-10 computer at MIT.

With a syntax resembling ALGOL 60, MACSYMA has capabilities for manipulating algebraic expressions involving constants, variables, and functions. The user can differentiate, integrate, take limits, solve equations, factor polynomials, expand functions in power series, plot curves, etc. A user can also manipulate lists, subscripted variables, and matrices with many of the usual operators. Only a few of the system's many capabilities are demonstrated here. ↑s stops printout.

1. To set TIP parameters and connect to the MATHLAB PDP-10:

| | |
|---|---|
| @r [CR] | *Reset TIP, terminal-dependent setup here.* |
| @e [SP] r [LF] | *"Echo remote", echo at remote HOST.* |
| @L [SP] 198 [LF] | *Connects to MATHLAB PDP-10* |
| LOGGER | |
| T R OPEN | |
| MIT MATHLAB PDP-10 | |
| ITS.761. DDT.460. | |
| 7. USERS | |
| | *You are now talking to DDT, ITS's top-level program. The time-sharing system is waiting for you to log in.* |

2. To login, type:

| | |
|---|---|
| :login [SP] iccxxx [CR] | *Login as ICCXXX, XXX being your initials. Typing mistakes can be corrected by hitting the RUBOUT or DEL key which causes the last character typed to be deleted and echoed.* |

3. To return to time-sharing level from a job such as MACSYMA, type a control-z (denoted "↑z"), the ITS attention character:

| | |
|---|---|
| ↑z | *A control character is typed by holding down the <CTRL> key while typing the specified character, in this case a "z".* |

4. To get status of users, type:

:listf [SP] tty [CR]

5. To print monitor commands:

:? [CR]

6. To get helpful information, type:

:help [CR]

7. To send a message concerning some bug or problem with MACSYMA, exit to the time-sharing system with control-z ("↑Z") and type:

↑z
:mail [SP] macsyma [SP] ... multi-line message ... ↑c

*Note that control-c ("↑c") ends message.*

8. To use MACSYMA, type:

:macsyma [CR]

*MACSYMA requests input by typing an input line label, like "(C1)" below.*

To work with the expression $(x+1)^3$, you can type it in by using FORTRAN-like syntax as follows:

(C1) (x+1)**3@@

A MACSYMA input line is usually terminated with an at sign ("@"). To get the TIP to send out an at sign ("@"), the user must type "@@" and this will be echoed as "@@@". Dollar-sign ("$") may be used to terminate a MACSYMA command when the user wishes display of the result to be suppressed. MACSYMA does not distinguish between upper and lower case characters. In the above case, evaluation and simplification are null operations. MACSYMA will come back with:

```
                                3
(D1)                      (X + 1)
(C2)
```

*Note that your expression is displayed in a two-dimensional notation comparable to that of a textbook. Your result is assigned a label, D1, which may be used in subsequent commands.*

*MACSYMA automatically labelled the next input line C2.*

Let us use one of the over one hundred commands available in MACSYMA, a command for expanding expressions. Commands are written in functional notation, as follows:

```
(C2) expand(d1)@@
                          3     2
(D2)                     X  + 3 X  + 3 X + 1
```

One of the first things you will want to learn is how to correct your input line. There are several possible methods. You may delete the last character typed by hitting the RUBOUT or DEL key once. Doing so will cause the deleted character to be echoed. Sometimes you just wish to start all over again. To do this type "??" (i.e., two question marks), which deletes the current line. Editing is a more complex facility than you will need at first. We will not enter into it here, so you might as well retype your command, taking care to avoid errors.

Let us consider a few additional commands and facilities. To differentiate an expression, use DIFF(expr,var). Here "expr" is the expression or its name, "var" is the variable with respect to which differentiation is to be performed.

```
(C3) sin(x)*cos(x)@@
(D3)                          COS(X)  SIN(X)
(C4) diff(%,x)@@               2         2
(D4)                        COS (X) - SIN (X)
```

*Note the use of percent sign ("%") in C4. The symbol "%" always represents the previous expression, in this case D3.*

To differentiate an expression twice, use DIFF(expr,var,2):

```
(C5) diff(d3,x,2)@@
(D5)                          -4 COS(X) SIN(X)
```

There are a number of ways for effecting a substitution of one expression for another inside of a third. For example:

```
(C6) z*%e**z@@
                                 Z
(D6)                          Z %E
(C7) d6,z=x**2@@
                                 2
                              X  %E
(D7)                                 X
```

An equivalent command is:

(C8) substitute(x**2,z,d6)@@

(D8) $X^2\ \%E^{X^2}$

*Note the order of arguments to SUBSTITUTE: Substitute the first for every occurrence of the second inside the third.*

We shall now consider some more linguistic facilities available in MACSYMA. To assign an expression to a variable use:

(C9) a:%@@

(D9) $X^2\%E^{X^2}$

*Note that the variable A will have the value* $X^2 e^{X^2}$.

Thus:

(C10) a+1@@

(D10) $X^2\%E^{X^2} + 1$

To define a function F(Z) to be SIN(Z)+1, use ":=" in typing:

(C11) f(z):=sin(z)**2 + 1@@

(D11) $F(Z) := SIN^2(Z) + 1$

(C12) f(x+1)@@

(D12) $SIN^2(X + 1) + 1$

Equations in MACSYMA are a particularly useful form of expression. To represent the equation $X^2 + 2X = Y^2$, use

(C13) x**2+2*x = y**2@@

(D13) $X^2 + 2X = Y^2$

One may add expressions to equations, multiply an equation by an expression, and add two equations together.

(C14) d13+1@@

(D14) $X^2 + 2X + 1 = Y^2 + 1$

The left-hand-side of an equation is obtainable by the function LHS. RHS obtains the right-hand-side.

(C15) lhs(%)@@

(D15) $X^2 + 2X + 1$

Equations are generated as intermediate results of MACSYMA's SOLVE command. For example:

(C16) x**2-1@@

(D16) $X^2 - 1$

(C17) solve(%,x)@@ *Solve the equation D16=0.*

SOLUTION

(E17) $X = -1$

(E18) $X = 1$

(D18) [E17, E18]

*The final result of SOLVE is a list of its intermediate solutions.*

SOLVE can, among other things, obtain closed form solutions to polynomials which can be factored into linear, quadratic, cubic or quartics over the integers.

To substitute one of the solutions into the original equation you can type:

```
(C19) d16,e17@@
(D19)                                  0
```

*Since E17 evaluates to the equation X = -1, the substitution is made into $X^2$ -1 and the result is simplified to zero, as expected.*

One sometimes wants an expression containing a sum which is unevaluated or unevaluatable. For instance:

```
(C20) 'sum(g(i),i,0,n)@@
```

(D20)

$$\sum_{I=0}^{N} G(I)$$

*Note the use of an undefined function G. G may be given a definition or substituted for at a later time. Also note the use of the quote symbol. The effect here is to prevent an attempt to evaluate the sum. In this case, however, the quote makes little difference since we would have obtained the same result had we not quoted because the upper limit, N, has not been assigned a value and the SUM cannot be carried out. MACSYMA considers the quoted and unquoted form of a function to represent its "noun" and "verb" forms, respectively. Most functions are verbs and will be evaluated.*

The trigonometric functions (e.g., SIN, COS) are nouns and normally do not evaluate, even if given numerical arguments.

Thus:

```
(C21) sin(1)@@
(D21)                                SIN(1)
```

To evaluate trigonometric functions with numeric arguments use a NUMER specification:

```
(C22) sin(1),numer@@
(D22)                              0.84147098
```

You now may wish to use MACSYMA on your own. Skimming the MACSYMA manual should be helpful.

9. To log out of the system, type:

```
↑Z
:logout [CR]
```

*Typing control-z gets you back out of MACSYMA.*

10. To close network connections:

```
@c [LF]
T R CLOSED
```

## BBN LIFE === HOST #69

LIFE is the mathematical game described in Scientific American, Volume 223, #4, October 1970. It was originated by the mathematician John Conway at Cambridge. It was coded by Ray Tomlinson at BBN. LIFE simulates a colony of organisms living on a 72x72 rectangular grid. Each point except for those on the edges, has 8 neighboring points: 4 horizontally and vertically, and 4 diagonally.

The rules of LIFE are:

1. Birth
   A new organism is created on an empty grid point if exactly 3 neighbors are adjacent to the grid point.
2. Death
   An organism dies of overcrowding if it has 4 or more neighbors.
   An organism dies of isolation if it has fewer than 2 neighbors.

   Deaths and births happen simultaneously.

The program requests an initial colony pattern from the user. This is input by typing for instance:

```
   * * * [CR]
 *     * [CR]
   * * * [CR]
```

Use asterisks, spaces, and carriage returns. The standard TENEX editing characters may be used to edit any input, i.e., control-A will delete the previous character, control-X deletes the lines, and control-R retypes the line. The pattern is terminated with an altmode (ESC).

Each successive generation will be typed out until one of three things happens:

1. The colony dies
2. A stable pattern is established
3. Any teletype key is pressed

At that point, the program requests another initial pattern.

To play LIFE, proceed as follows:

| | |
|---|---|
| @r [LF] | *Reset TIP, terminal-dependent setup here.* |
| @L [SP] 69 [LF] | *Connect to BBN, BBN herald is followed by at sign ("@") prompt.* |
| @login [SP] iccc [SP] iccc [SP] 11514 [CR] | *Log into BBN.* |
| @run [SP] <hacks>life [CR] | *Start the LIFE program.* |

Play LIFE as described above.

To stop playing LIFE, type:

| | |
|---|---|
| ↑C | *Type control-C to return to EXEC.* |
| @logout [CR] | *To log out of BBN.* |
| @c [LF] | *To disconnect TIP from BBN.* |

## UCLA-NMC SIGMA-7 === HOST #1

UCLA-NMC runs an experimental system called "SEX" on their XDS Sigma-7. SEX interacts line-at-a-time and requires local echoing at the TIP. Commands should be in upper case alphabetics.

1. To set TIP parameters and connect to UCLA-NMC:

```
@r [LF]                 Reset current TIP parameters, terminal-dependent setup here.
@t [SP] o [SP] L [LF]   TIP to "transmit on line-feed".
@i [SP] L [LF]          "Insert line-feed" after carriage-return.
@l [SP] L [LF]          Connect to UCLA-NMC, HOST #1.
LOGGER
T R OPEN
                        Note, word LOGIN typed by system.
```

2. To login to MASTER at UCLA-NMC:

```
LOGIN ICCC [CR]         Note, upper case must be used from here forward.
JOB STARTED             "!" is the prompt from MASTER.
```

3. To see who is on the system:

```
! WHO [CR]
002 WHO     STARTED
USER PORT
SEX!  16
FK     3
FK     2
TL     0
ICCC  27
```

4. To get back to master:

```
X [CR]
```

5. To use the toy question answering program, TIMMY, type:

```
! TIMMY [CR]
002 TIMMY     STARTED
MY NAME IS TIMMY THE TERMINAL, WHAT'S YOURS?
YOURNAME [CR]
PLEASED TO MEET YOU, YOURNAME   HAVE WE MET BEFORE?
NO [CR]
SORRY, BUT I HAVE A TERRIBLE MEMORY FOR NAMES.  ANYWAY, MY JOB IS TO ANSWER
YOUR QUESTIONS SO, ASK AWAY.
WHEN WILL THIS COMPUTER CRASH NEXT? [CR]
ABOUT 5 O'CLOCK.
.....                   Ask any number of your own questions of TIMMY.
GOODBY [CR]             Exit from TIMMY.
```

6. To use the calculator program, type:

```
! ABACUS [CR]
002 ABACUS     STARTED
CALCULATOR
DO YOU NEED INSTRUCTIONS?
YES [CR]
.....                          The HELP section offers instructions and formats for
                               interacting with a powerful desk calculator.
.....
FIXED POINT?
NO [CR]
AUTOMATIC SUBTOTAL?
YES [CR]
FIRST ENTRY
400.23 [CR]
INVALID OP CODE                We need an op code like '+'.
+ 400.23 [CR]                  Note that each op code like '+' requires a space (SP)
                               following it.
S    400.2300                  We requested automatic subtotal.
+ 200.234 [CR]
S    600.4639
* 3.23 [CR]
S   1939.4978
- 1200.213 [CR]
S    739.2849
A [CR]                         Store contents of accumulator into area A.
S   739.2849
+ 200.1 [CR]
s   939.3848
IA [CR]                        Display contents of area A.
=   739.2849
T [CR]                         Get current total.
T   939.3848
FIXED POINT?
X  CR                      Exit ABACUS, return to MASTER.
```

7. To exit MASTER:

```
! X [CR]
JOB KILLED
LOGIN
```

8. To disconnect from TIP:

```
@c [LF]
T R CLOSED
```

## SCHOLAR === HOST #69

SCHOLAR is a program to review a subject area with a student. For this example, the subject is the geography of South America. SCHOLAR will ask questions and evaluate the student's answers. It will answer questions for the student.

1. To connect to BBN-TENEX:

```
@r [LF]                    Reset the TIP, terminal-dependent setup here.
@L [SP] 69 [LF]            Cause TIP "Logger" to connect to BBN Tenex where SCHOLAR
                           will run.
```

2. To login to BBN-TENEX:

```
@ log [SP] iccc [SP] iccc [SP] 11514 [CR]   BBN's EXEC prompt character is at sign ("@").
                                            The second "iccc" is the password and will
                                            not print.
```

3. To start the SCHOLAR program:

```
@ run [SP] <warnock>scholar [CR]    Because SCHOLAR is still undergoing development,
                                    it is not a BBN subsystem, but rather a program
                                    kept in the directory of one of its
                                    developers, <WARNOCK>.

BBN LISP-10 07-12-72...
GOOD MORNING
←                                   SCHOLAR runs under BBN LISP whose prompt character is
                                    backarrow ("←").
```

4. To start SCHOLAR dialog, type:

```
← sysin( <warnock>scholar.sys)      SCHOLAR system parameters loaded into BBN LISP.
FOR INSTRUCTIONS, PLEASE TYPE 'INSTRUCTIONS(SHORT)' OR
'INSTRUCTIONS(LONG)'
( <WARNOCK)SCHOLAR>SYS;9)
←
← scholar()                         After (possibly) looking at instructions, start the SCHOLAR
                                    dialog.
HELLO, MY NAME IS SCHOLAR.  I WILL HELP YOU IN REVIEWING THE GEOGRAPHY OF
SOUTH AMERICA.  THIS IS ABC REGIONAL HIGH SCHOOL.  TODAY IS 24-OCT-72.

IF YOU HAVE CAREFULLY STUDIED YOUR INSTRUCTIONS, WE MAY BEGIN THE REVIEW NOW.

WHEN YOU ARE READY TO START, TYPE READY --- REMEMBER TO TERMINATE EACH LINE
OF INPUT WITH AN ASTERISK * FOLLOWED BY A CARRIAGE RETURN.
* ready* [CR]                       Asterisk ("*") is SCHOLAR's prompt character.  You must,
                                    repeat MUST, terminate input to SCHOLAR with an asterisk
                                    ("*") and a carriage return before it will listen to you.
                                    Use control-a ("↑A") to delete characters from your input
                                    line, i.e., the standard Tenex editing characters.
PLEASE TYPE YOUR NAME---MR., MISS, or MRS. FOLLOWED BY FIRST NAME, MIDDLE NAME
OR MIDDLE INITIAL, AND LAST NAME.
* YOURNAME* [CR]
NAME OF YOUR INSTRUCTOR, PLEASE:
* warnock* [CR]
```

```
LET'S BEGIN OUR DISCUSSION NOW, "YOURNAME"
USE ONE OF THE FOLLOWING:
14400000
3600000
1200000
1800000
TO ANSWER THE QUESTION:
APPROX WHAT IS THE AREA OF BRAZIL?
* 14400000* [CR]
WRONG,
I'LL GIVE YOU ANOTHER CHANCE.
* 3600000* [CR]
VERY GOOD.
.....
```

*Continue your dialog as you wish. When tired, type control-c ("↑C") to return to EXEC to logout.*

```
.....
↑C
```

5. To logout:

```
@ logout [CR]
```

6. To disconnect from BBN-TENEX:

```
@c [LF]
```

UCLA-CCN 360/91 TSO === HOST #65

The UCLA Campus Computing Network (CCN) 360/91 offers a number of services to the ARPANET including IBM's Time-Sharing Option (TSO). TSO is an interactive programming system sitting on top of the awesome power of an IBM 360/91.

The CCN 360/91 assumes line-at-a-time ARPANET interaction. Both upper and lower case input are allowed. CANcel or control-X ("↑X") deletes the current line and Backspace or control-H ("↑H") deletes the previous character on the same line.

1. To set TIP parameters and connect to CCN:

```
@t [SP]o [SP] L [LF]      "Transmit on linefeed", TIP sends CCN a line at a time.
@i [SP] L [LF]            "Insert linefeed" after every carriage return.
@L [SP] 65 [LF]           Cause TIP "Logger" to connect to CCN 360/91.
LOGGER                    @ E L [LF]
T OPEN R OPEN
UCLA CCN 360/91 SERVER TELNET.
VERSION 2.5 30 APR 1972
ENTER COMMAND OR 'HELP':
```

2. To get help:

```
help [CR]
COMMANDS AVAILABLE ARE:
SERVICE--DESCRIPTION
RJS......EBCDIC REMOTE JOB SUBMITTAL SERVICE.
ARJS.....ASCII REMOTE JOB SUBMITTAL SERVICE.
TTYRJS...ALTERNATE ASCII RJS FOR A MODEL 33 TTY.
BBOARD...BULLETIN BOARD NOTICES OF GENERAL INTEREST
TSO......ACCESS TO IBM TSO TIME SHARING SYSTEM.
HELP.....PRODUCES THIS INFORMATION.

COMMANDS HAVE NO OPERANDS BUT MUST BE FOLLOWED BY
A CR/LF.  ANY NONAMBIGUOUS ABBREVIATION FOR A COMMAND
IS ACCEPTABLE.  FOR FURTHER INFORMATION ABOUT CCN
SERVICES, CALL (213)825-7548.

ENTER COMMAND OR 'HELP':
```

3. To get current system schedule:

```
bboard [CR]
#1036 - 1 AUG 72 - 07.28.50 - OPR
CCN HARDWARE AND SOFTWARE MAINTENANCE SCHEDULE FOR 1972-73:
(FACILITY CLOSED FROM 07:00 AM SUNDAY TO 08:00 AM MONDAY)
.................
ENTER COMMAND OR 'HELP':
```

4. To use TSO:

```
tso [CR]
WELCOME TO UCLA/CCN TSO
IKJ54012A ENTER LOGON -    TSO message requesting you to log in; "!" is the TSO prompt.
```

5. To login to TSO:

```
! logon [SP] icc [CR]        User name "icc". If already in use, use "icx", where x is any
                             digit from 1 to 9.
LOGON ICC
ENTER PASSWORD
! iccc [CR]                  Your password is "iccc".
ICC LOGON IN PROGRESS AT 07:38:14 ON SEPTEMBER 15, 1972
WELCOME TO TSO.  TSO IS AVAILABLE FROM 0600 to 1400 PDST.
ENTER 'NEWS' FOR CCN NEWS
READY
```

6. To request help:

```
! help [CR]
.................
```

7. To instruct TIP to "send synch" to get TSO's attention, equivalent to TTY BREAK or 2741 ATTN:

```
! @s [SP] s [LF]
READY
```

8. To get current time:

```
! time [CR]
CPU - 00:00:02  EXECUTION - 00:00:38  SESSION - 00:05:51
READY
```

9. To see who is on the system:

```
! users [CR]
3 USERS
USER     UNIT
WDD      (06D)
AKB      (046)
HCL      (042)
READY
```

10. To send a message to another user:

```
! send [SP] 'Hello, this is a user at ICCC' [SP] user(wdd) [CR]
READY
```

11. To use edit to create a FORTRAN program:

```
! edit [SP] sqrtXXX [SP] new [SP] fortg [CR]        XXX should be your initials.
INPUT                        Edit automatically puts numbers on lines.
000010! 1 [SP][SP][SP][SP][SP] format (' enter a ') [CR]   Use spaces, e.g., 'f' is the
                                                           7th character.
000020! 2 [SP][SP][SP][SP][SP] format (' the square root of a is ',f10.3) [CR]
000030! 3 [SP][SP][SP][SP][SP] format (f10.3) [CR]
000040! 100 [SP][SP][SP] write(6,1) [CR]
000050!  [SP][SP][SP][SP][SP][SP] read(5,3) a [CR]
```

```
000060: [SP][SP][SP][SP][SP][SP] b=sqrt (a) [CR]
000070: [SP][SP][SP][SP][SP][SP] write(6,2) b [CR]
000080: [SP][SP][SP][SP][SP][SP] go to 999 [CR]
000090: [SP][SP][SP][SP][SP][SP] end [CR]
000100: [CR]                                    Blank line takes you from input
                                                to edit mode.
EDIT
! change [SP] 80 [SP] /999/100/ [CR]    To correct statement error in GOTO statement
                                        number 80.

! list [CR]
00010  1    FORMAT(' ENTER A')
00020  2    FORMAT(' THE SQUARE ROOT OF A IS ',F10.3)
00030  3    FORMAT(F10.3)
00040  100  WRITE(6,1)
00050       READ(5,3) A
00060       B=SQRT(A)
00070       WRITE(6,2) B
00080       GO TO 100
00090       END
END OF DATA
```

12. To save the text:

```
! save [CR]
SAVED
```

13. To exit edit:

```
! end [CR]                    You can type "help edit" if you have problems in EDIT.
READY
```

14. To change the attention getting character:

```
! terminal [SP] input($) [CR]          Makes "$" the attention getting character.
READY
```

15. To compile program just written in edit:

```
! fortg [SP] sqrtxxx [CR]
BEGIN COMPILATION
FORTRAN IV G LEVEL  20   MAIN     DATE = 72259   12/09/21
.................
                        Compilation listing here.
```

16. To load and run program:

```
! gofort [CR]                    Loader listing here.  Program enters execution.
ENTER A
! 56.2 [CR]
THE SQUARE ROOT OF A IS       7.497
ENTER A
! 64.0 [CR]
THE SQUARE ROOT OF A IS       8.000
ENTER A                        The program is observed to work.
! $ [CR]                       Get back to command level with attention character defined
                               above.
READY
```

17. To display user catalogued data sets:

```
! listcat [CR]
SQRTXXX
HELLO
READY
```

18. To delete a data set:

```
! delete [SP] sqrtXXX [CR]
READY
```

19. To log out of TSO:

```
! logoff [CR]
ICC LOGGED OFF TSO AT 12:59:33 ON SEPTEMBER 15, 1972+
```

20. To log out of ccn:

```
! logoff [CR]
IDJ54012A ENTER LOGON -
```

21. To disconnect from TIP:

```
! @c [LF]
T R CLOSED
```

BBN Chess === HOST #69

CHESS is the chess-playing program developed by Richard Greenblatt at MIT. It was described in "The Greenblatt Chess Program" at the 1967 Fall Joint Computer Conference. The program is an honorary member of the United States Chess Federation and the Massachusetts Chess Association, under the name Mac Hack Six. In the April 1967 amateur tournament, the program won the class D trophy; it wins about 80% of its games against non-tournament players.

During play, the program understands moves typed in using standard chess notation, some examples of which are given below.

| | |
|---|---|
| P-KN3 | Pawn to king's knight 3 |
| B*P | Bishop captures pawn |
| O-O | Castle kingside |
| QR-Q1 | Queen's rook to queen 1 |
| R/K2-Q2 | Rook on king 2 to queen 2 |
| P-R8 | Promote pawn (to queen assumed) |
| Q*P/Q6 | Queen captures pawn on queen 6 |
| O-O-O | Castle queenside |

Other commands are available for control and information:

| | |
|---|---|
| BD | Type out board |
| PW | Play white |
| PB | Play black |
| PN | Play neither |
| PS | Play self (both sides) |
| M | Make next move |
| U | Undo last move |
| DRAW | Request machine to acknowledge draw |
| PG | Print game (history)) |
| LIST | List commands |
| RESET | Overturn board (for bad sports) |

1. To prepare the TIP and connect to BBN TENEX, type:

   @r [LF] — *Reset TIP, terminal-dependent setup here.*

   @L [SP] 69 [LF] — *Cause TIP "Logger" to connect you to BBN Tenex, HOST #69.*

2. To login to BBN Tenex, type:

   @ login [SP] iccc [SP] iccc [SP] 11514 [CR]

   *The Tenex EXEC prompt character is "@". The second "iccc" is your password and will not print. In the Tenex EXEC and most subsystems, typing control-a ("↑A") will delete the last character typed on the current input line. Typing control-c will return you to the Tenex EXEC.*

3. To run CHESS, type:

   @ run [SP] <hacks>chess [CR] — *Run the CHESS program out of the directory <HACKS>.*

   ← pb [CR] — *The CHESS prompt character is "←". Tell CHESS to "Play Black", giving you the first move.*

```
← p-kb3 [CR]                    Your move (in this example) is "pawn to king bishop 3" as
                                indicated in standard chess notation.

← B P/K2-K4                     CHESS makes its answering move, "pawn from king 2 to king 4".
                                Your turn again.

bd [CR]                         You request to see the board (BD).

WR  WN  WB  WK  WQ  WB  WN  WR
WP  WP  **  WP  WP  WP  WP  WP
--  **  WP  **  --  **  --  **
**  --  **  --  **  --  **  --
--  **  --  BP  --  **  --  **
**  --  **  --  **  --  **  --
BP  BP  BP  **  BP  BP  BP  BP
BR  BN  BB  BK  BQ  BB  BN  BR

← p-kn4 [CR]                    It is still your move (after typing out board) so you enter
                                "pawn to king night 4".

←B Q/Q1-KR5 CHECKMATE           CHESS puts you into checkmate, oops, you lose.

pg [CR]                         You request a summary of the game.

1  P/KB2-KB3      P/K2-K4
2  P/KN2-KN4      Q/Q1-KR5

reset [CR]                      You reset the board to play another game.
```

When finished, type control-c ("↑C") to get out of CHESS and return to the Tenex EXEC to log out.

```
↑C
```

4. To logout of the BBN TENEX system, type:

```
@ logout [CR]
```

5. To disconnect, type:

```
@ c [LF]
T R CLOSED
```

MIT-DMCG MUDDLE === HOST #70

The MIT Project MAC Dynamic Modelling and Computer Graphics (DMCG) PDP-10 runs the ITS time-sharing system developed at the MIT Artificial Intelligence Laboratory.

ITS prefers to do its own echoing, a character at a time. Its attention getting character is control-z ("↑Z"). Typing DEL or RUBOUT will generally delete the last character typed on input. Control-g will generally abort commands. To suppress output, type control-s. At command level, upper and lower case alphabetics are treated alike.

1. To set TIP parameters and connect to MIT-DMCG:

```
@r [LF]                 Reset the TIP, terminal-dependent setup here.
@e [SP] r [LF]          "Echo remote", DMCG ITS prefers to do its own echoing.
@L [SP] 70 [LF]         Cause TIP "Logger" to connect you to DMCG ITS, HOST #70.
MIT PROJECT MAC DMCG PDP-10.
PLEASE LOGIN WITH YOUR HOST NUMBER FOLLOWED BY
YOUR INITIALS (E.G., BY TYPING "LOGIN &)RMM").
MONIT.192
.....
                        A system message of the day will appear here.  It can be
                        suppressed by typing control-s ("↑S").
```

2. To login to MIT-DMCG:

```
; login [SP] iccXXX [CR]     Login as "iccXXX" where "XXX" is your initials.
```

MUDDLE is an interpreter related to the list processing language LISP. It improves on LISP. It improves on LISP in a number of ways including its general treatment of data types.

3. To invoke the MUDDLE interpreter:

```
; muddle [CR]
MUDDLE 31 IN OPERATION
.....
                        MUDDLE mail is typed here.
LISTENING-AT-LEVEL 1 PROCESS 1
```

The canonical first step in learning a language like MUDDLE is to define the function FACTORIAL, rescursively. MUDDLE type-in is terminated by typing ESCAPE (ALTMODE). The following multi-line function definition is a single MUDDLE transaction which we have broken into lines for explanatory purposes. Note carefully the use of upper case (MUDDLE distinguishes upper and lower case) and the delimiting spaces in MUDDLE type-in which we have not emphasized in our usual bold way.

```
<DEFINE FACT (N) [CR]
```

Define FACT to be a function of one variable, named "N" inside of FACT. In typing its definition (to follow), use DEL or RUBOUT to delete an incorrectly typed character and control-L ("↑L") to get the current input buffer typed out. Try control-L.

```
<COND [CR]
```

FACT is a simple CONDitional with the following clauses.

(<Ø? .N> 1) [CR]

If the argument to FACT is zero (0?), then FACT returns the value 1, as expected.

(ELSE <* .N <FACT <- .N 1>>>) >> [ESC]

If the argument to FACT is not equal to zero (ELSE), then FACT returns the product (*) of the value of N and the value of FACT applied to N minus 1. Here we balance the brackets for the product (given in prefix notation as is MUDDLE's normal mode), balance the parenthesis of the second conditional (ELSE) clause, and balance the brackets for the COND and DEFINE, respectively. With the function definition complete, we type ESCAPE to get MUDDLE to process it.

FACT

Having evaluated our definition of FACT and entered it in its memory, MUDDLE returns the function name FACT.

<FACT 5> [ESC]

Try FACT out on a few small numbers whose FACTORIAL you know; like 5 for example, whose FACTORIAL is 120 (5x4x3x2x1x1).

120

MUDDLE performs the specified evaluation and returns the correct value.

<FACT 13> [ESC]

Try FACT for 13 now; a more challenging computation.

6227020800

MUDDLE returns the correct answer, as you can verify with pencil and paper in a few long minutes.

<FACT 14> [ESC]

Now try 14, a slightly larger number for FACT.

*ERROR*

OVERFLOW

*

LISTENING-AT-LEVEL 2 PROCESS 1

MUDDLE encounters a number too large to represent as an integer in the PDP-10. The overflow condition, detected during an invocation of the function "*" (multiply), signals an error condition. Because we have made no provision for this condition ourselves, it is passed untrapped up to the user, you.

<ARGS <FRAME <FRAME>>> [ESC]

The environment in which the error occurred remains for user perusal at this higher ERROR level. You examine the arguments to the function "*" (multiply) which caused it to overflow. The arguments are found in a push-down-stack frame.

[14 6227020800]

It must be that 14 times 6227020800 is too large an integer for the PDP-10 to represent.

<ERRET> [ESC]

Understanding the difficulty, you do an error return (ERRET) to back out to the top level, to try something new. The error environment, having supplied you with what you needed to know, is now discarded.

LISTENING-AT-LEVEL 1 PROCESS 1

<FACT 33.0> [ESC]

Try FACT with a floating point number.

.86833170E37

That seems to work with the precision of a 36-bit word floating point number (7.2 decimal digits).

<FACT 34.0> [ESC]

```
*ERROR*
OVERFLOW
*
LISTENING-AT-LEVEL 2 PROCESS 1
<ARGS <FRAME <FRAME>>> [ESC]
```

Again we have overflow in the function "*" (multiply). Again we look at the offending arguments on the preserved runtime stack.

```
[34.000000 .86833170E37]
<ERRET> [ESC]
```

We now know the limits of our simple recursive definition of FACTORIAL using single precision PDP-10 arithmetic. Return to MUDDLE's top level by doing an error return (ERRET).

```
LISTENING-AT-LEVEL 1 PROCESS 1
↑Z                          Type control-z ("↑Z") to return to the DMCG ITS MONITor.
```

4. To logout of the DMCG system:

```
; logout [CR]
@c [LF]                     Close the TIP connection.
```

UCLA-NMC HELP === HOST #1

HELP is a subsystem at UCLA-NMC which permits a user to interrogate a database which is organized in directed graph form. Each vertex of the graph has a paragraph of information, including some information about further details which can be obtained from vertices which are reachable from the current one.

Thus, the user moves from vertex to vertex, investigating each item as his interest directs.

1. Setting up the TIP to talk to UCLA-NMC. Type the following:

| | |
|---|---|
| | *Note: LF means linefeed; CR means carriage return.* |
| @r [LF] | *Resets the TIP, terminal dependent set-up follows.* |
| @t [SP] o [SP] L [LF] | *Sets up TIP to transmit on linefeed.* |
| @i [SP] L [LF] | *Causes LF insertion after any CR.* |
| @L [SP] 1 [LF] | *Sets up connection to UCLA-NMC.* |
| LOGGER | *TIP says you have the LOGGER.* |
| T R OPEN | *TIP says connection is open.* |

2. Logging in to UCLA-NMC:

When the connection is open, the SEX timesharing system at NMC will type 'LOGIN' at you. If it does not, type a couple of carriage returns at it.

The following actions should get you logged in:

| | |
|---|---|
| LOGIN iccc [CR] | |
| JOB STARTED | *There may be a long delay between the 'Job Started' message and the '!' prompt character. Also, at this point, some message may be typed at you from the system. Wait for them to finish. The '!' prompt character means you are talking to the monitor called MASTER. You can instruct MASTER to start and stop programs, log you out, etc. Let's start the HELP program.* |
| ! | |

3. Using the HELP system

To start the HELP system, type as follows:

| | |
|---|---|
| ! help [CR] | |
| NNN HELP STARTED | *Note: NNN is the process number assigned to HELP; it is not of much concern to you.* |
| DO YOU KNOW HOW TO USE THIS PROGRAM? | |
| no [CR] | *If you say 'no', you'll be given a tutorial on the use of HELP. Eventually, you will wind up with the following prompt:* |
| ENTER A SERVICE NAME, X, OR ? | *At this point you should type ? to get a list of things you can get help about.* |

? [CR]

THE FOLLOWING HELP FUNCTIONS ARE AVAILABLE: (IN ALPHABETICAL ORDER)

HELP...*short description given here*

LOGIN...*ditto*

MSG...*how to use our message processor*

NETWORK...*tutorials on network resources*

SRVYGRAPH...

TELNET...

SURVEY... ...*some random comments...*

```
ENTER A SERVICE NAME, X, OR ?        You can now type one of the names listed
                                     above to get information on any of those
                                     programs, or 'X' to terminate the HELP
                                     subsystem.  Try typing in NETWORK [CR] or
                                     SRVYGRAPH [CR].
                                 If you type 'X', the following happens:
x [CR]
!                                You have returned to MASTER.
                                 At this point, you can try some of the programs you read
                                 about in HELP, or you can logout of SEX by typing 'X' again.
x [CR]
JOB KILLED
LOGIN
```

If you have left a program running when you logout, you will see the following:

```
x [CR]
LOGOUT?                          If you want to go back to MASTER, type 'iccc [CR]'.
                                 If you want to logout, type 'logout [CR]'.
logout [CR]
JOB KILLED
LOGIN
```

5. Closing the connection to UCLA-NMC:

   Type the following:

```
@c [LF]                          Tell TIP to close connection.
T R CLOSED                       Transmit and receive connections are closed.
```

BBN DOCTOR === HOST #69

DOCTOR is a LISP program written by Joseph Weizenbaum and described in "ELIZA - A Computer Program For the Study of Natural Language Communication Between Man And Machine" in the Communications of the ACM, January 1966.

DOCTOR simulates a psychiatric interview with a Rogerian psychotherapist. The 'subject' types in some statement in natural language using normal punctuation and sentence structure. Two editing characters are provided: control-A deletes the previous character typed (repeatable back to the previous space or comma), and number sign ("#") deletes the entire line typed in. Each input statement is terminated by a period or question mark. The session is terminated by typing GOODBYE.

1. To connect to BBN TENEX:

   @r [LF] *Reset the TIP, terminal-dependent setup here.*

   @L [SP] 69 [CR] *Connect to BBN.*

2. To login to BBN TENEX:

   @ login [SP] iccc [SP] iccc [SP] 11514 [CR] *Note that the password (i.e., the second "iccc") will not be echoed.*

3. To run DOCTOR:

   @ run [SP] <hacks>doctor [CR]

   BBN LISP-10 02-22-71 ...

   GOOD AFTERNOON.

   TELL ME YOUR PROBLEMS.

   PLEASE TERMINATE INPUT WITH A PERIOD OR A QUESTION MARK.

   * i have a big problem. *The user begins as in a psychiatric interview.*

   I AM NOT SURE I UNDERSTAND YOU FULLY. *The DOCTOR continues the interview in a lively fashion.*

   * my grandmother has a gzorenplatz.

   IS IT IMPORTANT TO YOU THAT YOUR GRANDMOTHER HAS A GZORENPLATZ?

   .....

   *Continue dialog as your spirit carries you.*

   .....

   * goodbye.

   IT'S BEEN MY PLEASURE, THAT'S $7.15 PLEASE.

   NIL

   ←

When finished, type control-c, logout, and disconnect.

↑C

@ logout [CR]

.....

@c [LF]

T R CLOSED

SAIL PARRY === HOST #11

PARRY is a program which simulates a paranoid patient in a mental hospital. The program was written by Prof. Ken Colby of Stanford University. The user of this program must play psychiatrist and attempt to discover the patient's problems without antagonizing the patient or scaring him so much that he runs away.

Initially, the user is asked to select some behavior parameters for the patient to be modelled. Once this has been done, you are on your own with a psychotic (paranoid) patient sitting at the other end of your teletype!

The attention getting character is control-c ("↑C"). Typing ↑C will return you to the system executive. To delete the last character typed on input, type the character DEL or RUBOUT. The system's executive command interpreter prompts with period (".").

1. Setting up the TIP to talk to SAIL:

| | |
|---|---|
| @e [SP] r [LF] | *sets up remote echo* |
| @i [SP] L [LF] | *inserts Linefeed after each carriage return* |
| @L [SP] 11 [LF] | *initiates connection to SAIL* |
| LOGGER | *TIP says you are being connected* |
| T R OPEN | *TIP says you are connected* |

2. Logging in to SAIL:

After you have been connected to SAIL by the TIP, you must log in. SAIL may type out a number of messages at you before you can login, so be patient. Striking control-c will stop the printing. A typical interaction is shown below.

```
SAIL PDP-10.
PLEASE LOGIN AS "NET,GUE".NOTE:  NCP IS STILL BEING DEBUGGED
.................
L NET,GUE  .login [SP] IC,CC [CR]
```

You may get a message back saying "Are you sure?" at this point. If this happens, just YES [CR].

```
JOB 27 STANFORD 6.09B 9-19-72
FRIDAY 22-SEP-720944
```

At this point, many messages may come out. Be patient.

```
↑C                          system leaves message processor automatically
.
```

You are now logged in to SAIL; the "." is the executive system prompt character.

3. How to run the PARRY program:

When the system types ".", you are talking to the executive system. You can now ask for the PARRY program. An example is given below.

```
. r[SP] parry [CR]

SUPPRESS NON VERBAL FEATURE?  (Y,N)      End input parameters with carriage return.
* y [CR]
VERSION (WEAK,STRONG)
* weak [CR]
.................
    Continue specifying parameters of interaction.
.................
```

```
TRACE VARIABLES? (Y,N)
* n [CR]
ARE TWO TELETYPES BEING USED? (Y,N)
* n [CR]
DO YOU WANT THIS INTERVIEW SAVED ON A FILE? (Y,N)
* n [CR]
END INPUT WITH A PERIOD OR QUESTION MARK, FOLLOWED BY TWO ALTMODES.
SPELL OUT NUMBERS.
TO INDICATE SILENCE, TYPE 'S.'
WHEN FINISHED, TYPE 'BYE.'
READY:
*                          At this point, your patient is waiting for your response.
```

The following is a portion of an exchange between a sane(?) programmer and his psychotic patient:

```
* good [SP] morning. [ESC] [ESC] $*$     Note: the [ESC] [ESC] is typed by the programmer,
                                          and the system responds with '$*$'. The '$' is
                                          the echo for ESC.
HELLO
READY:
* who are you? [ESC] [ESC] $$
WHAT DO YOU MEAN?
READY:
* what is your name? [ESC] [ESC] $$
MY NAME IS HAROLD SMITH
READY:
* how are you feeling today, harold? [ESC] [ESC] $$
ALL RIGHT I SUPPOSE
READY:
* tell me your problems. [ESC] [ESC] $$
PEOPLE GET ON MY NERVES SOMETIMES
.....
                           To leave the PARRY program, you must type control-c ("↑C").
```

4. How to logout of SAIL:

```
↑C                         To logout, first be sure you are talking to the executive
                           program by typing "↑C".
. kjob [CR]                "Kill" your job.
JOB 28, [IC,CC] LOGGED OFF TTY 122 10:50 22 SEP-72
.11 HOURS, CONSOLE TIME
.08 MINUTES, CPU TIME
.40 K AVERAGE CORE
KJOB
```

5. How to close the SAIL connection:

```
@c [LF]
T R CLOSED
```