R. W. Berner

Speeches and Papers

3

1955-1960

ASSOCIATION FOR COMPUTING MACHINERY
211 East 43rd Street
New York 17, New York

VISITING SCIENTISTS PROGRAM

FOR COLLEGES AND UNIVERSITIES

1963

Committee:

Dr. James R. Oliver, Director
USL Computing Center
University of Southwestern
Louisiana
Lafayette, Louisiana, 70506

Dr. Jack Moshman, Vice-President
C-E-I-R, Inc.
Arlington, Virginia

Dr. John W. Hamblen, Director
Data Processing and Computing
Center
Southern Illinois University
Carbondale, Illinois

BRUCE W. ARDEN, University of Michigan, Computing Center, Ann Arbor, Michigan. Topics of Interest: (1) On the Classification of Languages and Compiling Algorithms; (2) The Use of Computers in a University.

ROBERT L. ASHENHURST, School of Business, University of Chicago, Chicago 37, Illinois. Topics of Interest: (1) Switching Theory (especially decomposition theory); (2) Computer Design and Organization; (3) Computer Arithemtic (algorithms and behavior); (4) Formal Structure of Algorithms (numeric and non-numeric).

ROBERT W. BEMER, Director Systems Programming, UNIVAC Division of Sperry Rand Corporation, 315 Park Avenue South, New York 10, New York. Topics of Interest: (1) Numerical Analysis; (2) Programming Languages; (3) Automatic Control.

HOWARD BROMBERG, C-E-I-R, Inc., Benson East, Jenkintown, Pennsylvania, 19046. Topics of Interest: (1) Automatic Programming; (2) Computer Languages; (3) Standardization Principles.

JOHN W. CARR, III, Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia 4, Pennsylvania. Topics of Interest: (1) Languages for Problem Solving Using Digital Computers; (2) Better Computers in the Future; (3) Numerical Analysis and Functional Analysis; (4) Automatic Theorem Proving.

THOMAS E. CHEATHAM, JR., 1 Ardley Place, Winchester, Massachusetts. Topics of Interest: (1) Trends in Programming Systems; (2) The CL-II Programming System.

CHARLES H. DAVIDSON, Department of Electrical Engineering, University of Wisconsin, Madison 6, Wisconsin. Topics of Interest: (1) Organizing and Operating Computing Centers; (2) Teaching of Programming to Engineering Classes; (3) Open-Shop Vs. Closed-Shop Computing Center Operation.

BERNARD A. GALLER, Computing Center, 1000 North University Bldg., University of Michigan, Ann Arbor, Michigan. Topics of Interest: (1) The Organization of University Computing Center; (2) Trends in Automatic Programming; (3) Defining New Operations Into a Procedure-Oriented Language.

RICHARD A. HAMMING, Consultant, Head, Mathematical Systems Planning Department, Bell Telephone Laboratories, Murray Hill, New Jersey. Topics of Interest: (1) Numerical Analysis and Information Theory; (2) Use of Computers in Other Fields; (3) Think Big Even With a Small Computer; (4) Numerical Analysis Vs. Mathematics.

VISITING SCIENTIST PROGRAM

Through financial assistance by the National Science Foundation, the Association for Computing Machinery is pleased to announce a program of Visiting Scientists in the field of computing. Under the terms of the grant, a total of forty visits will be made by scientists who have distinguished themselves in this field to colleges and universities in the United States. Western Michigan University has been selected as one of the forty universities to participate in the Visiting Scientist Program.

We take great pleasure, therefore, in announcing the visit of:

ROBERT W. BEMER

Director, Systems Programming

UNIVAC Division of Sperry Rand Corporation

to

Western Michigan University

Monday, May 6, 1963:

Room 170, Wood Hall - 8:00 P.M.

"COMPUTERS" (Open to the Public)

Tuesday, May 7, 1963:

Mathematics Department Colloquium
Room 371, Wood Hall - 4:00 P.M.

"NUMERICAL ANALYSIS AND COMPUTERS"

Coffee will be served in Room 338 at 3:30 P.M.

Robert W. Bemer has served in the following capacities:

1949-51 - Employed by the RAND Corporation as a programmer.

1951-52 - Lockheed Aircraft Company, as a Group Leader in Mathematical Analysis.

1952-54 - Marquardt Aircraft Company, as Manager, Numerical Analysis Group.

1954-55 - Lockheed Aircraft Corporation, Missiles and Space Division, as Manager of a Mathematical Analysis Group.

1955-62 IBM Corporation, as Director of Programming Standards.

1962- UNIVAC Division of Sperry Rand Corporation, as Director of Systems Programming.

/S/ Jack R. Meagher, Director
Computer Center

SPURIOUS COMPUTATIONAL ACCURACY

INTRODUCTION

THIS SPURIOUS ACCURACY OFTEN STEMS FROM PRECISION AVAILABLE. DEFINE.
IMPROPER USAGE OF COMPUTERS LEADS TO THIS. ENGINEERS AND OTHERS OFTEN TOO GULLIBLE, READ 8 DIGIT ACCURACY INTO NUMBERS RESULTING FROM CALC ON THREE DIGIT INPUT.
TOUGHER THAN IN OLD DAYS. MACHINES SO MUCH FASTER SEEMS MIRACULOUS, LEADS TO UNWARRANTED CONFIDENCE. IMPORTANT THAT EVERY USER REALIZE.
DIFFERENCE BETWEEN EXPECTED/TOLERATED ERRORS AND UNEXPECTED/ UNFORESEEN/ UNSUSPECTED ERRORS.

1. FLOATING POINT ARITHEMTIC (OR AUTO SCALING)

HOW DONE IN THE VARIOUS COMPUTERS, HARDWARE OR SOFTWARE, STANDARDIZATION (NOT NORMALIZE, NOT GENERAL ENOGUH). VARIATIONS IN RANGE OF EXPONENT, LENGTH OF FIXED POINT PART, FUNCTION OF COMPUTER WORD SIZE (BINARY) OR SUBROUTINE (DECIMAL CHARACTER).
ACTUAL MECHANICS OF MULTIPLY, THEN ADD.
PROBLEM WITH EXACT ZERO AND COMPUTED RELATIVE ZERO. DIVIDE BY ZERO OR RELATIVE ZERO. BUILDING IN TOLERANCE.

```
                                        0000XXXX0 XXXX
    x  zeros on right indicate the problm,    XXXX XXXX
       beware!                          0000 XXXX 0000
```

OVERFLOW AND UNDERFLOW OF EXPONENT. OPTIONS OF REPLACEMENT OR STOP FOR ANALYSIS. MUST BE PROGRAMMED OR SWITCH SET. APPLY EXPECTED LIMITS TO VARIABLES, ALSO BY PROGRAMMING, CATCH SOONER.
NON NORMALIZED F.P. ARITHMETIC. VARIOUS METHODS. METROPOLIS MACHINE AT CHICAGO, MANIAC III. OTHERS, SUCH AS STRETCH. EVEN SEMI NORMALIZED AS INBETWEEN SOLUTION. GRAUS METHOD TO MATCH REAL VARIABLES, BOTH LEADING AND TRAILING ZEROS.
VARIOUS METHODS OF ROUNDING IF NOT BUILT IN. NOISY MODE, RANDOM , NOT ALWAYS BY XXX .5

2. CURVE-FITTING OF EMPIRICAL DATA @ 2 or 3 DIMENSIONS

USUALY DONE BY POLYNOMIALS RATHER THAN TABLES FOR REDUCTION IN STROAGE REQUIREMENTS AND SPEED.
VERY SIMPLE TO DO BY COMPUTER, EQUAL AND UNEQUAL INTERVALS, AUTOMATICALLY DERIVE ALL ORDERS AND SELECT FIRST TO MEET TOLERANCE REQUIREMENTS. WATCH OUT, FOR HIGHER MAY GET WORSE.
PARTITION RATHER THAN GO TO HIGHER THAN 5 or 6 DEGREE IN REAL WORLD. ALSO, LOGIC BEATS COMPUTATIONAL TIME. WORD HERE ON HASTINGS, DOES NOT USE LOGIC, THEREFORE INTRODUCES ADDITIONAL ERROR.
AVOID USING CONTINUED FRACTIONS FOR UNEXPECTED DISCONTINUITIES, UNLESS ROUTING AUTOMATICALLY DETECTS.
DANGER OF USING TOO MANY DIGITS IN COEFFICIENTS. TELESCOPE. ALSO TELE-SCOPE TERMS TO REDUCE NUMBER OF COMPUATIONS, THEREFORE ALSO REDUCE ERROR.

3. ERRORS IN SUCCESSIVE COMPUTATIONS

FOR REAL SUCCESS HAVE TO RUN COINCIDENTAL ERROR ANALYSIS. OFTEN TURNS OUT TO BEE TOO IMPRACTICAL. DO BY RULE OF THUMB.
EXAMPLE FROM DR. CONWAY, * 8 digit ending up as effectively 5. IN SOME CASES PEOPLE SAY DONT EVEN ROUND, ITS BETTER. OK TO CARRY EXTRA SIGNIFICANCE IN ITERMEDIATE CALC, BUT BE SURE TO TAKE IT OFF AT END!
DONT FORGET ROUNDING GOES IN OPPOSITE DIRECTIONS FOR X + and - SIGNS, AND THE DISTRIBUTIPON MAY NOT BE UNIFORM.
OFTEN GOOD TO DUPLICATE CALCULATIONS, SINCE COMPUTERS ARE SO XXXX FAST. DO ROUNDED AND UNROUNDED. IN BINARY USE NOISY MODE, THAT IS , FILL WITH ONES INSTEAD OF ZEROS ON FLOATING POINT SHIFTS, AND SEE WHAT THE DIFFERENCE IN RESULT IS.
EXAMPLE OF XXX 500 x 500 MATRIX INVERSION WITH 27 bits of FPP. MAY HAVE TO GO TO TRIPLE PRECISION FOR MEANING IN MOST CASES. HOWEVER ACTUALLY DID WORK IN SPECIAL CASES OF SPARSE MATRICES. REINVERT AND COMPARE. BY ITERATION CAN WORK FOR A WHILE IN SINGLE PRECISION AND THEN GEARSHIFT FOR SLOWER MULTIPRECISION.

4. RELATIVE VS. ABSOLUTE ACCURACY

EXAMPLE OF SIN X APPROX X. IS IT NECESSARY TO USE RELATIVE ACCURACY IN THE REAL WORLD. NOT USUALLY. ABSOLUTE IS MOSTLY PREFFERED FOR COMPUTER USAGE. SEEN A LOT OF WASTED EFFORT AND COMPUTER TIME FIGGHTING THIS.

5. CASE EXAMPLE

VARIABLE STEP METHODS FOR ADAMS AND RUNGE-KUTTA METHOD INTRODUCED IN 1951. USUALLY SUFFICES TO HALVE OR DOUBLE STEP SIZE. ANY OTHER METHOD CAUSES TOO MUCH COMPUTATIONAL WORK. HALVE IF OVER MAX ALLOWABLE ERROR, DOUBLE IF UNDER MINIMUM REQUIRED. DIFFERENCE EQUATIONS TO DIFFERENTIAL, MODIFY, BACK TO DIFFERENCES.
EXAMPLE OF TOUGH CASE NEAR ORIGIN IN TRAJECTORY COMPUTATION. EXTREME SINGULARITY. AUTOMATIC STEP ADJUSTER HALVED AND HALVED TO 2^{-36}, which is LOWEST BIT POSSIBLE ON MACHINE. GAVE RESULTS TO MATCH PHYSICAL DATA. NO OTHER METHOD COULD DO.

6. TABEE PRESENTATION AND OUTPUT

often advisable to PRESENT ANSWERS WITH BUILTIN RANGE TOLERANCE TO SIGNIFY POSSIBLE INACCURACY, SINCE IT MAY VARY WITH THE PARAMETER COMBINATIONS. OK ON SOME TABLES WITH CONSTANT ERROR BOUND, BUT NOT USUAL. XX 32-36 sorting time, e.g.

COMPUTERS

~~PLEASURE TO RETURN TO MICHIGAN, LIVED HERE AS YOUTH.~~
~~LIKE GENERALITY OF TITLE, GIVES MUCH FREEDOM. PARTICULARLY~~ WISH TO DISCUSS IMPACT OF COMPUTERS UPON MANS WORKING WORLD.

SHORT HISTORY

KEEP IN TRUE PERSPECTIVE AS A DEVICE FOR AUXILIARY POWER.
ALWAYS IN GOOD REPUTE WITH MATHEMATICIANS, PASCAL DEVISED A MACHINE TO HELP DAD WITH THE SHOP, NAPIERS BONES, BABBAGE, COUNTESS LOVE LACE. PI TO XXX 500 plus PLACES, LIFETIME WORK, FOUND ERROR IN 260th or so.
MODERN ORIGINS 42-46 ECKERT MAUCHLY ENIAC, VON NEUMANN PRINCETON + JOHNNIAC. UNIVAC, IBM XXX 70], ETC. FROM MECHANICAL TO ELEC TRONIC, VACUUM TUBE TO TRANSISTOR, CARD TO MAG TAPE, DRUMS, DISCS, PHOTO PLATES, CRYOGENICS, TUNNEL DIODES, WIRES (NOT OLD DELAY LINES),
GROWTH SIZE OF STORAGE. L950 typical about 50 characters or digits, now 30,000,000 per drum, many units, tapes at about]2,000,000 characters, or over 2 million words, avg govt imstallation 18 tapes. SPEED 5 PER SEC IN]950, 2,500,000 in 1960, ? for 65? EVEN INVENT NEW WORD FOR IT NANOSECOND. THOUSANDTH OF A MILLIONTH, IDENTICAL TO US BILLIONTH.

WHAT IS A COMPUTER

DIFFICULT TO GET A GOOD DEFINITION. IFIP WORKING ON NOW. BASICALLY A DEVICE CAPABLE OF MANIPULATING, TRANSFORMING AND COMMUNICATING DATA WHICH REPRESENTS INFORMATION.
THE BIT AS A BASIC UNIT OF INFORMATION INFORMATION THEORY. ALWAYS WORK IN BITS. OFF ON, LIKE LIGHTS. GROUPS OF BITS REPRESENT SYMBOLS. LOOK AT TELETYPE PUNCHED TAPE, E.G. A COMPUTER MANIPULATES THESE SYMBOLS, NOT JUST ARITHMETIC. THIS IS A SUBSET ACCORDING TO CERTAIN RULES. THATS HOW GOT A START, BUT MUCH BRANCHED OUT INTO NONNUMERIC WORK NOW, AS EVIDENCED BY THE BIG CITY TELEPHONE DIRECTORIES.
DIFFERENCE IN SYMBOL MANIPULATION BY MAN / COMPUTER. FASTER IN PRESET MODE, NOT AS VERSATILE TO CHANGE PROGRM AS YET. MORE ADAPTABILITY, MULTIPROCESSING IN PARALLEL REQUIRED. WIFE HARRY ? THOUGHT OF CAREY, NOT SAME ACTOR. THEN HARY MCDONALD, NO. NEXT PROGRAM MAC DONALD CAREY, IMMEDIATLE FREED BLOCK , LED TO HARRY DAVENPORT. C COMPUTERS MAY NOT BE ABLE TO DO THIS FOR MANY YEARS, IF XXX EVER.
WORK ON SYMBOL STRINGS IN ONE DIMENSION. RECURSIVE PROPERTIES OF CHARACTERS. EVEN IN CHINESE PICTURE SYMBOL. TWO SYMBOLS FOR FEMALE WITH A BROKEN LINE OVER = ~~TROUBLE~~, FOR TWO WOMEN UNDER ONE ROOF IS ALWAYS TROUBLE.
HOW TO DO ARITHMETIC? ALWAYS TALK OF BINARY, FISTS. ACTUALLY USED IN OCTAL, WITHOUT THUMBS. DUODECIAML FAD OF SOME YEARS BEFORE.

WHAT IS PROGRAMMING

INSTRUCTIONS BUILT INTO HARDWARE. STORED PROGRAM, CAN OPERATE UPON SELF. THIS IS THE BASIS DEVELOPMENT. CAN LOOP, MAKE BRANCHES ON DECISIONS, GENERALLY REPRODUCE ANY COMPUTABLE FUNCTION _ SEE TURING MACHINE (NOT A COMPUTER ON TOUR)

CAN PROGRAM IN LANGUAGE OF MACHINE, BUT HOW TERRIBLY COMPLICATED!

PROGRAMMING IS BASICALLY THE EQUIVALENT OF EDUCATION. WE EDUCATE IT TO PERFORM CERTAINS TASKS. LEVEL OF EDUCATION VARIES. CUSTOMER OR USER GIVES EQUIVALENT OF ON THE JOB TRAINING. THIS IS OFTEN TEDIOUS ON REPETITIVE OR SIMILAR TASKS. WE NOW EQUIP WITH GENERALIZED EDUCATION EQUIV TO COLLEGE. MAHINE UNDERSTANDS CERTAIN LANGUAGES OF FORMAL, RIGOROUS AND STYLIZED FORM. FORTRAN, ALGOL, COBOL. ADVAN TAGES OF COMMON LANGUAGES FOR INTERCHANGE, EASIER LEARNING. ETC. NORTH AMERICAN HAS 3000 ENGINEERS USING F. MORE EFFICIENCY, COSTS OF GETTING COMPUTER TO DO USEFUL WORK. SAGE AT 50, OTHERS at 5-10, F AT A DOLLAR OR LESS.

SIZE OF PROFRESSION. IS IT A GOOD ONE? INTERDISCIPLINARY. WHAT ARE FUTURE PROSPECTS AS INDIVIDUAL PROFESSION OF INFORMATION PROC ESSING? WHO QUALIFIES? TRAINING IN UNIVERSITIES, MIT BASE MAND_ ATORY. HELPS LOGICAL THINKING, AUXILIARY DEVICE FOR LEARNING.

INTERMEDIATE EDUCATION OF TRADE SCHOOL TYPE. NOW COMING INTO PROMINENCE. PRESET AND SELECTIVE. CUT SNIP PASTE AND SPEC_ IALIZE.

ADVANCEMENT IN PROGRAMMING

FOUND PROBLEMS IN VARIATIONS OF CODES. BLOCKS INTERCOMMUN ICATION. RECENT ASA STANDARD, REPLACES BAUDOT, FIELD ATA, ETC. 8 BIT CODE INSTEAD OF 5, ATT ALL SET UP, 11th DIGIT IN DIRECT DIALING. EUROPE ALSO ADOPTING. GREAT FOR SATELLITE COMMUN ICATIONS, INFLUENCES ENGLISH AS THE INTERNATIONAL LANGUAGE, FOR ECONOMIC REASONS, NOT SUPERIROITY. NEW EQUIPMENT TO BE BUILT DIRECTLY TO CODE. INTERMEDIATE PHASE OUT PERIOD. ADVANTAGES IN STRUCTURE

CAN MACHINES THINK? OLD FASHIONED CONTROVERSY, NEW TWIST IN ALGORIT HMIC VERSUS HEURISTIC. (SHOW TRIANGLE FOR HEURISTIC) TAUBE WROTE BOOK TO DEBUNK IN AVAILABLE TIME. HOWEVER, RUSSIANS EXPEND PRIOR ITY ENERGY ON ASSUMPTION THAT CAN DO. UNSOLVABLE NOW DUE TO INAB ILITY TO PREDICT DEVELOPMENTS WHICH MAY MAKE IT POSSIBLE. CHESS, CHECKERS, ETC. AD IN SCIENTIFIC AMERICAN, US CHESS CHAMPION. BISGUIER THEN NOBLE, FRENCH RIVIERA. NEEDLESS TO SAY, I DIDNOT PLAY.

SOME APPLICATIONS, OLD AND NEW

MANY OLD NOW GETTING TO BE WELL KNOWN. START IN ACCOUTING, NATURALLY. SCIENTIFIC AREA, AIRCRAFT, MISSILES, EVEN WEATHER PREDICTION IS OLD HAT NOW.

LANGUAGE TRANSLATION IS OLD BUT UNSUCCESSFUL SO FAR. AMBIGUITIES, AS NATURAL LANGUAGES HAVE SYNTACTIC, SEMANTIC AND PRAGMATIC CAPABILITIES FOR CONTENT.NOT DECIDABLE AND RIGOROUS. MANY EXAMPLES. CLASSIC IS OUT OF SIGHT, OUT OF MIND TO BLIND AND INSANE WHEN RETRANSLATED. RUSSIAN BOOK ON ALGO TRANSLATED MECHANICALLY BY ibm ON AIR FORCE STUDY. CALLED INPUT LANGUAGE, CAME OUT ENTRANCE TONGUE. TAKE THE AMBIGUITY IN " A PRETTY LITTLE GIRLS SCHOOL" ++++++ DIFFICULT TO GET EVEN FROM CONTEXT. SPOKEN IS BETTER FROM GESTURES, INFLECTION, VOLUME ETC. MAY BE BETTER IF PRIOR REQUIREMENTS FOR TRANSLATION ARE KNOWN, CAN GIVE AUXILIARY INFO.

NEW APPLICATIONS IN MEDICINE, BIO ELECTRONICS. HOSPITAL CLINIC AND TESTING CONTROL, DIAGNOSTIC AIDS, RECORDKEEPING. JUST TALK FOUR YEARS AGO, NOW MEDICAL CONFERENCES ON COMPUTER WORK. ALSO IN LAW. LAYMAN ALLENS WORK AT YALE LAW SCHOOL. WFF AND PRROF GAME. WELL FORMED FORMULAE. TEACH SYMBOLIC LOGIC BY PROGRAMMED INSTRUCTION. MULL, MECHANICAL USAGE OF LOGIC IN LAW, QUARTERLY, SURPRISED TO FIND.

EDUCATION GENERALLY, PROGRAMMED LEARNING, TUTOR TEXTS TRANSFORMED TO COMPUTER CONTROL, MAY EVENTUALLY GET FULL FEEDBACK WITH COMBINATION DIGITAL TV TAPE AND COMPUTERS. CERTAINLY NEED SOMETHING LIKE THIS. OUR TEACHING SEGMENT OF THE POPULATION IS NOT FROM THE MOST CAPABLE GROUP, DUE TO LONGTIME ABUSES. MUST MAXIMIZE GOOD TEACHING TALENT.

LIBRARY WORK, INFORMATION RETRIEVAL, CONTROL OF LIBRARY AS AT NOTRE DAME, KWIC INDEX, DESCRIPTOS, CATALOGING. IMPORTANT AT LEAST 90 % OF US RESEARCH WORK IS PROBABLY DUPLICATION. SOME IS GOOD BUT NOT THIS MUCH

ONLINE CONTROL OF MANUFACTURING PROCESSES. REALTIME REQUIREMENTS NOW COMING TO FORE. BUT MOST COMPUTERS NOW DO NOT HAVE THESE CAPABILITIES. PRODUCTION CONTROL AND SCHEDULING.

COMMUNICATIONS NETWORKS FOR DATA COLLECTION, PROCESSING, SEE INETR RELATIONSHIPS. INTERNAL REVENUE IS CERTAINLY DOING IT. HOWEVER, SUCH TECHNIQUESHAVE A MORE CHEERFUL SIDE, FOR WE CAN THUS OPTIMIZE OUR EFFORTS. MAY ENABLE US TO BEAT RUSSIA .

STORE & FORWARD-WESTINGHOUSE, AIRLINES RESERVATION & CONTROL

HOW FUTURE CONPUTERS WILL WORK

COMMUNICATIONS LINKAGES, VIA SATELLITE. CENTERS, I/O STATIONS , NOT LKE MIT WITH TYPEWRITERS, BUT PRODUCTION I/O DEVICES SO THE USER CAN BELIEVE HE HAS THE BIG MACHINE. PRO RATA. PRIORITY, COMMUTATION. DISCONNECT PERIPHERAL DEVICES FOR SEPARATE WORK, IF DESIRABLE.

GREAT EXPANSION OF MARKET AND THEREFORE UTILIZATION. ALMOST EVERYONE MAY BE EXPECTED TO BE AS FAMILIAR WITH COMPUTERS IN TEN YEARS AS THEY ARE WITH THEIR AUTOMOBILE NOW. OF COURSE, I ADMIT I DONT KNOW MUCH ABOUT CARS. I HAVE SPECIALIZED.

WHAT IS THE UNIVERSITY ROLE?

SOURCE OF MOST ORIGINAL COMPUTER WORK. THE ACTUAL HARDWARE DESIGN HAS FADED, BUT WORK IN SOFTWARE (DEFINE THIS) HAS REPLACED IT.

WORK OF MIT , CASE, BERKELEY IN REMOTE I/O. LIGHT PENS. APT FOR NUMERICAL TOOL CONTROL, WIDESPREAD NOW. MUCH LANGUAGE TRANSLATION WORK.

NEW FIELD OF COMPUTER ASSISTED DESIGN, MECH DRAWING.

TRAINING THE WELL ROUNDED AND LOGICAL MIND, NOT NECESSARILY FAMILIARITY WITH PROGRAMMING COMPUTERS, BUT AS A TOOL IN FASTER ASSIMILATION AND PROPER CLASSIFICATION AND ARRANGMENT OF THE INDIVIDUALS KNOWLEDGE.

NEED FOR NEW COURSES IN PROBLEM SOLVING LOGIC, SYNTHESIS.

EFFECTS OF AUTOMATION

CONTROVERSIAL RIGHT NOW. ADMIT THAT PRESS HAS NOT BEEN HANDLES CORR ECTLY, SO THAT PEOPLE CAN FEAR COMPUTERS. HOWEVER! WHO DROVE HERE TONIGHT? THINK IS A GOOD ANALOGY, BUT DONT WORRY. LOOK AT THE HORESHOEING CYCLE. AUTOS INVENTED, TRADE DIED OUT. TV INVENTED, RESURGENCE IN THE HORSESHOEING BUSINESS FOR WESTERNS. SUPPLY AND DEMAND STILL WORKS. ADJUSTMENTS ARE ALWAYS NECESSARY. FORTUNATELY, ALTHO COMPUTERS CARRY THE SEEDS OF THEIR OWN DESTRUCTION, THEY ALSO ARE A MECHANISM FOR THEIR OWN SALVATION. THAT IS, THE ABILITY TO MASS XXXXXX RETRAIN PEOPLE FOR BETTER AND MORE EXCITING, DIGNIFIED WORK. EDISON SAID INVENTION WAS 10% INSPIRATION, 90% PERSPRATION. THE IDEA IS TO USE THE COMPUTER TO DO THE DIRTY PERSPRING WORK

I THINK COMPUTERS CAN IMPROVE PEOPLE. THERE IS NO SUCH THING AS A XXX FIXED I.Q. (OCONNOR WIGGLY BLOCK). THERE IS MUCH SPARE CAPACITY IN THE HUMAN MIND. THE TRICK IS TO GET IT LINKED UP IN AN EFFICIENT FASHION. WE KNOW THE PATHS CAN BE RESTRUCTURED. THE MOST EXCITING PROFESSION OF THE NEXT DECADE MAY WELL BE THAT OF A COMBINATION BRAIN XXXXXXXXXX AND COMPUTER SPECIALIST. FORTUNATELY THERE DO NOT SEEM TO BE ANY RELIGIOUS ARGUMENTS AGAINST SUCH IMPROVEMENT OF MAN: MEDICAL REPAIR OF THE BODY HAS SET THE PRECEDENT. REPAIR AND IMPROVEMENT OF THE MIND WILL NOW HAVE THE AID OF A POWERFUL TOOL -- THE COMPUTER.

INSERT BY 1600 AIRLINES/BRIDGES. STILL ETHICS! FOR CHOICE

January 10, 1963

Mr. James P. Anderson, Supervisor
System Design Section
Systems Department
Burroughs Corporation
Research Center
Box 843
Paoli, Pennsylvania

COBOL vs. MACHINE LANGUAGE — THAT MOOTLESS CONTROVERSY

Dear Mr. Anderson:

As far as I can see, I do not have anything scheduled in March that might conflict, so this is open to your decision.

As a summary for the title already furnished:

> "Countless hours of debate have been wasted pondering the selection of a language. Should or should not one accept object program inefficiency for future compatibility? As so often happens, the optimum is neither yes nor no, but a blend of both. This arises from the proper use of the ENTER verb, not into assembly language but rather into precompiled subroutines of the type that caused the heavy usage FORTRAN II now enjoys."

Cordially yours,

R. W. Bemer, Director
Systems Programming

RWB:rs

ACRONYM
NEGATIONS
LOW BIDR.

(IF THING CAN GO FOR 200 ! LOST WARES)

COBOL VS. ML — THIS NEEDLESS CONTROVERSY (TITLE, SEX)

A LITTLE HISTORY — US STEEL CREATION SCENE

SUGGESTED A MIXED BAG TO PREVENT SABOTAGE

SWINGING TO COBOL, THEN BACK IN ALMOST REVULSION WHY?

PURPOSES — INTERCHANGE
UNDERSTANDING
MAINTENANCE
FAST CONVERSION
STANDARDS

COMPARISON WITH FORTRAN.

FIRST GENERATION, EXPERIENCE — ANY TEST . DOD PRESSURE, LACK OF CONFIDENCE.

CONTROVERSY? — ℵ , BOTH COMPILE & RUNTIMES.

REQ, OPT, ADDED

(MODULAR) CONSTRUCTION OF THE PROCESSOR — CORE VARIATION — STR TOOLS FOR ℵ IN SIZE, MAINT

TOTAL SYSTEM — FII PRECOMP, NOW C, SII , USE OBJ AM INTERCHANGEABLY.

ADVANTAGES OF MORE STANDARDIZATION IN SYSTEM AS WELL AS LANGUAGES.

HOW TO USE PROPERLY

PHASING IN — DO IN PARALLEL — C/F FIRST THEN CHIP AWAY FOR ℵ , DO IN SUBR, . REDO

KEEP IN PARALLEL — CAN SERVE AS FLOW CHART — QUOTE McCRACKEN. GOOD PRACTICE!

ℵ — McCRACKEN IN DEFINITION, COBOL BOOK — OVER 490 176 → 88 → 46 SEES, MUST STILL BE PROGRAMMERS.

DIAGNOSTICS — HEAVY ON THIS — USS EXPERIENCE . FIV — AIRFORCE. 9% DIAGRAMS 35/65. WHAT COULD YOU DO IF THIS WORKED? SIMPLE "SEARCH, DELETE, ETC — SET THEORY
MULTIPLE BUGS PER COMPILE — NON STOP. — QUICK MODIF OF SOURCE STMT!

POWERFUL ASSEMBLY — HIT PROCEDURES ! LIKE "DEFINE" WON'T WORK.

INDENTED ELEMENTS DO — MAINTAIN SIMILARITY FOR AUTO TRANSLATION!

IN PRACTICE — PL/I , UP (FIV), ETC. IF OTHERS TO USE, DO FRAMEWORK IF MUST.

SO ITS USELESS ! USE JUDICIOUS SELECTION OF ALL!

MINN-STP ~ OCT 62

DRAFT

Increased Processor Flexibility Thru Better Design

Without Increased Costs

Specialization for the Individual User Thru Modular Construction -

(1) Philosophy - Can be done (with proper planning) at $0 and better.

- Individual elements said not worth the cost, but many!
- In business to help customer run EFFICIENTLY. Should not saddle with unused extras, intricate intermarriage of routines (makes for bad maintenance). But must balance modularity and h - it could get out of hand in extreme cases.

(2) Variations in Processor Design

(A) Core size- Careful tabul of affected places. Build on small, run large or vice versa?

Indirect to table - Plug in.

computation each time.

This is reason for S/OBJ environment in COBOL - Don't just ignore it. However, do not go under minimum (709 8K/32K), std config per program, as 1050

(B) Unused components - RDR, PUNCH - Don't carry dead weight -

Reassemble for h, expand table sizes and limits - control this by assem control cards to strip out A to B.

This demands no interlocking, use access to common. In case of concurrency, add obj run space, e.g., an installation might make several versions of varying power.

Importance of subroutine linkage conventions and adherence.

(C) Swap or Tradeoff Components - Drum↔Tape, Cards↔Tape, etc.

Commonality thru std. interface, just like HDWE for I/O-

Does not know source, only rate-which may be buffered.

Always use image and conventions.

(D) Master flow plan - Who calls on whom in what seq? VITAL FOR MODIFI-cation on site. Need tech. docum, prog. listing - profusely annotated, normally customer does at own risk but could be different, depending on built in facility or not.

(E) Importance of Mockup. - Simulate in time, then replace with working component -, can't do unless modular, important to get going. Big payoff in easier modification.
Also possible to process interpretively, then replace. Can do in part or whole. Also for testing ≡ replaceable or interchangeable parts.

INTRO.- Know I have chosen a tough topic for tonite; one that needs doing- ~~whether or not you think my job precludes my understanding this type of detail~~.

Nasty job- because involves telling most Systems Programmers they haven't been doing the job right from a utilitarian, customer viewpoint.

~~Don't be suspicious of anything I propose (you non-UNIVAC people, that is). I'm here in a professional capacity - ACM, and the secrets are free tonite.~~

Variation in Language Design

Changes - Not basically sound, for Stds & compat, reasons. User who does goes on own at his risk. Must be strongly cautioned that facilities (introduced for other purposes) must not be subverted lightly.

Superstructures - Yes - Shadow, others → Fortran and Algol. Use building blocks for special functions.

Subsets - Often OK to strip out Lang facilities (like recursive, own, complicated editing I/O, etc.) to yield subsets which process faster. However, also at own risk.....

The London School of Economics and Political Science

(University of London)

Houghton Street, Aldwych
London, W.C.2
Telephone HOLborn 7686

25th April 1962

R. W. Bemer, Esq.,
Thomas J. Watson Research Centre,
P.O. Box 218,
Yorktown Heights,
New York, N. Y., U. S. A.

Dear Bob,

Yes! Another letter from me! — COULD NOT, NEW UNIVAC POSITION

This time I am writing to you to ask you to participate in the programming symposium which I am organising. Would you care to accept an invitation to be the guest of honour at our dinner and to talk on a general topic such as "Programming System Development in the U.S.A."?

I am afraid that we would not be in a position to pay your fare. However, you might be able to justify your trip by saying that the conference would give you a bird's-eye-picture of the state of the programming art in Great Britain. If you were able to come for the whole week you would probably have a privileged "big gun" status, and I am sure that your comments would help to let in a little wholesome American "fresh air" on the British scene.

If you cannot manage yourself, perhaps you could suggest someone else who would be suitable.

Enclosed is a further copy of the programme.

Yours sincerely,

Peter

Peter Wegner

THE LONDON SCHOOL OF ECONOMICS AND POLITICAL SCIENCE
(UNIVERSITY OF LONDON)

Telephone: HOLBORN 7686
Telegrams: POLECONICS, ESTRAND, LONDON

HOUGHTON STREET,
ALDWYCH,
LONDON, W.C.2.

PROGRAMMING SYSTEMS SYMPOSIUM

JULY 2 - 6

In response to a number of requests from industry, a symposium on programming systems is being planned at the London School of Economics in the week July 2 - 6.

The basic aim of the symposium is an expository one, and talks have been arranged so as to provide a graduated introduction to systems programming for the relatively unsophisticated programmer. However, it is hoped that the symposium will also be of interest to sophisticated systems programmers. It is hoped to review in some detail a number of existing programming languages and programming systems, to discuss techniques of implementation, and to consider future trends. Talks will normally last one hour and will be followed by thirty minutes of discussion.

PRELIMINARY PROGRAMME

Monday

9.30 a.m.	S. Gill	Introductory Talk
11.15	P. Wegner	Examples from FORTRAN, Algol and Cobol
2.00 p.m.	I. C. Pyle	Introduction to FORTRAN
3.45	M. Woodger	Introduction to Algol
5.30	Cocktail Party	

Evening 7.30 - P. Wegner: The Burroughs Compiler Game

Tuesday

9.30 a.m.	P. Wegner	Intermediate Languages and Programming Systems
11.15	I. C. Pyle	Atlas FORTAN Implementation
2.00 p.m.	B. Randell	Algol Implementation
3.00	G. H. Huxtable	" "
4.15	C. A. R. Hoare	" "

Evening 7.30 - Discussion on Compilers: Woodger, Pyle, Randell, Huxtable, Hoare.

Wednesday

9.30 a.m.	J. K. Iliffe	Addressing and Automatic Storage Allocation Techniques
11.15	D. Morris	A Compiler Compiler for the Atlas
2.00 p.m.	A. D'Agapeyeff	Commercial Compilers
3.45	J. Harwell	FACT

Evening 7.30 - E. Humby - Rapidwrite. Followed by Discussion of Commercial Compilers.

Thursday

9.30 a.m.	S. Gill	Introduction to Timesharing
11.15	D. Howarth	A Supervisory System for the Atlas
2.00 p.m.	Speaker to be announced	Time Sharing Aspects of the Stretch Computer System
3.45	C. Strachey	Programming in Functional Notation
6.30	Dinner - Guest of Honour to be announced	

Friday

9.30 a.m.	Speaker to be announced	Principles and Applications of List Processing
11.15	J. K. Iliffe	Continuous Evaluation
2.00 p.m.	W. Burge	GIPSY - A General Interpretive Programming System
3.45	Discussion on Programming Systems	

IF YOU ARE INTERESTED IN RECEIVING FURTHER DETAILS REGARDING THE SYMPOSIUM, PLEASE COMMUNICATE WITH ME AT THE ABOVE ADDRESS.

The fee for the symposium, including lunches, morning coffee, afternoon tea, cocktail party and Thursday dinner will be £15.

Peter Wegner
16 April 1962

PLEASE POST ON NOTICE BOARD

THE PRUDENTIAL ASSURANCE COMPANY LIMITED.
(INCORPORATED IN ENGLAND.)

HOLBORN BARS, LONDON, E.C.1.

COMMUNICATIONS ON THE SUBJECT OF THIS LETTER SHOULD BE ADDRESSED TO "THE GENERAL MANAGER"

TELEGRAMS: "PRUDASCO CENT. LONDON."
CABLES: "PRUDASCO LONDON."
TELEPHONE: HOLBORN 9222.

OUR REF. MR/1/ELW.
YOUR REF.

30th March, 1962.

R.W. Bemer, Esq.,
International Business Machines Corp.,
112 East Post Road,
White Plains,
New York.

Dear Mr. Bemer,

Your letter of 9th February to Richard Goodman has been shown to me, as one of the authors of "Some Commercial Autocodes".

My colleagues and I are very aware that the Comparative Study, which was begun as an experiment during a brief lull between two EDP projects, covers only a small part of the field of study; there were several other areas of comparison which we were very keen to tackle. However, our computer is now imminent, and there is no prospect of our being able to take the study any further for the time being. We are delighted that you are intending to extend what we believe to be a much needed piece of work and should be most interested to see anything you publish.

If you should happen to be coming to London at any time I should be most honoured if you would care to address the British Computer Society study group of which I am Chairman. Its official title is "Advanced Programming" but our main interest for the past three years has been the problems of Autocode languages from both the user's and the compiler writer's point of view. Please accept a standing invitation to come and see us if ever you find yourself in our area.

Yours sincerely,

E.L. Willey.

IBM SYSTEMS RESEARCH INSTITUTE

Lecture Announcement

Time: Tuesday, February 20, 1962
7 - 8 p.m.

Place: 787 United Nations Plaza (at 44th Street)
Fifth Floor Auditorium

Speaker: Mr. R. W. Bemer
Director of Programming Standards

Topic: "International Standardization and Information Processing"

Abstract: Standards activities affecting the computer and information processing industries will be surveyed with particular reference to the impact upon computer users. Some insight will be given on how national and international standards are achieved from a complicated interlocking of professional organizations, business organizations, and government.

Standards in various stages of completion will be described from which it may be seen that standards negotiations are usually as difficult and thankless as those on mutual disarmament. Examples will be given of proposed standards which have been rejected because of their narrowness and limitations.

All IBM employees engaged in technical activities in the metropolitan area are invited to attend. The IBM Systems Research Institute sponsors frequent lectures in topics of interest in digital computer systems studies. These are usually scheduled for Tuesday evenings.

February 15, 1962

DENVER 1961

The subject for today is Computers and Education. I should like to borrow a technique from one of the speakers here today, Dr. Perlis, who remarked that one of the proposed papers for the 1959 International Conference in Paris could provide enough for an entire conference by simple permutations of the title. The title, as I recall it, was something on the order of "Quantitative Potential in Research Methods." Dr. Perlis pointed out that there were equally good papers possible under the titles

"Quantitative Research In Potential Methods"
"Quantitative Methods of Potential Research"
etc., etc., etc.

It seems to me that we could make similar permutations here today. We could have humans educating computers about humans, humans educating computers about computers, computers educating humans about computers, computers educating computers, and we could even have computers educating humans about humans educating computers, etc., etc. All of these are perfectly valid areas which may be mechanized, subject to the indispensible educational ingredient of feedback.

Many things have been said to carry within them the seeds of their own destruction. It could be said about computers that they carry within them the seeds of their own salvation. You are all aware of the automation furor in this country today.

Actually automation itself is not necessarily destructive of jobs. I admit that horseshoeing was an unprofitable business when an invention of one level came along -- that is the automobile, yet an invention on a higher level, TV, made horseshoeing profitable again because of the demand for Westerns.

We will all agree that it is good for the human to have the perspiration work (as Edison phrased it) performed by a machine and to have the inspirational work performed by the human. The only dispossession I have seen from computer automation is due to the inability of the present educational system to reeducate the displaced human to a higher and more intellectual job. Fortunately the same contrivance that displaces these humans is capable of reeducating them, thus hopefully insuring against its own extinction by a maddened mob.

Words are abstract symbols. They are brought to the human consciousness when:

(a) pictorialized in letters.

(b) pictorialized in non-LETTER symbols.

(c) enunciation in sounds

(d) other possible methods, such as telepathy?

Thus the written or spoken word does not signify the English language, which has a syntactical structure in meaning rules quite apart from the way the individual words are communicated between persons. Given an English or lunar equivalent of every German word it would be possible to speak German, although the average German would not recognize it as such.

23 January 1961
Corporate Logical Systems Standards

"Survey of Modern Programming Techniques"

Mr. L.A. Langreich

A copy of my paper to the British Computer Society is attached. It has received quite a favorable review in the December 1960 issue of the Computer Bulletin (published by the British Computer Society), and will be printed in its entirety in the March '61 issue. It has received favorable notice in this country through distribution to CODASYL. Requests for copies have been made by RCA, General Electric, etc. Mr. Lerner of DP Information also evinced interest.

Knowing the natural reservation of the British, I consider it no small honor to be the first American chosen to address the Annual General Meeting of the Society.

R.W. Bemer,
Manager, Corporate
Logical Systems Standards

RWB/jes

Attachment.

- - - - - - - - - - - - - - - - Advanced Notice - - - - - - - - - - - - - - - -

ACM-SIAM

Announcement of Meeting

Tuesday Evening, November 22, 1960

The Villa Chartier

Just South of Hillsdale Ave on El Camino Real

San Mateo, Calif.

We have been very fortunate to obtain Mr. R. W. "Bob" Bemer, Manager of Logical Systems Standards for the IBM Cooperation as guest speaker for the evening. Bob will cover the general topic of Problem Oriented Languages, with emphasis on a survey of the common elements of these languages and a report on where IBM now stands in the development of some of them. Bob is flying out from White Plains, New York for the express purpose of addressing this meeting.

Because of the widespread interest in this timely subject, and because Bob has many friends in the area, a very large turn-out is expected. We have reserved the Garden and Patio Rooms of the Villa Chartier for the evening. We can only accept reservations up to the limit of the capacity of these rooms. Please make your reservations early by calling Herb Finnie or Miss Viramontes at REgent 9-4321, ext. 28344.

160
PAV. MAX
=75

6:00 p.m.	Social Hour - A special bar will be set up in the Patio or Garden Room to accomodate members, their guests, and visitors. Please do not stop in the main bar, but come directly to the appointed rooms.
7:30 p.m.	Dinner - Turkey with all the trimmings, served in the famous Villa Chartier tradition--$3.75, includes the tax, tip, and speaker's dinner.
8:30 p.m.	Speaker: Mr. R. W. Bemer

Departing from some of our past practices, a table will be set up at the entrance to the meeting room where you can drop your $3.75, pick up your meal ticket, and be checked off on the reservation list. The tickets will be collected when you are served. It is hoped that this procedure will eliminate some of the confusion and disturbance we have experienced in the past.

MAKE YOUR RESERVATIONS EARLY!!!
Non-members are Welcome

THE AMERICAN UNIVERSITY
School of Government and Public Administration

Seventh Institute on Electronics
in Management
October 31-November 4, 1960

Monday, October 31

9:15 a.m. Welcome

Catheryn Seckler-Hudson, Dean
School of Government and Public Administration

Announcements

Lowell H. Hattery, Director
Center for Technology and Administration

THEME: ADP EQUIPMENT

9:30 a.m. What's Available in ADP Equipment

W. Howard Gammon
Data Systems Research Staff
Office Secretary of Defense

10:45 a.m. What's Ahead in ADP Equipment

S. W. Alexander
Chief, Data Processing Systems Division
National Bureau of Standards

1:30 p.m. How to Select among Computers

Lt. Colonel Harry F. Sieber, Jr.
Chief, Automatic Data Processing Branch
Systems Planning Division
Office of the Chief Signal Officer

3:45 p.m. Magnetic Tape for Data Storage

Walter L. Anderson
Vice President
General Kinetics, Inc.

and

Robert P. Gutterman
Vice President
General Kinetics, Inc.

Tuesday, November 1

THEME: TALKING TO THE COMPUTER

9:00 a.m. Automatic Scanning - Census and FOSDIC

James L. McPherson
Machine Development Officer
Bureau of the Census

10:30 a.m. Data Input from Automatic Instrumentation

Henry L. Mason
Supervisory General Engineer
National Bureau of Standards

1:30 p.m. Common Languages for Computers

R. W. Bemer
IBM Corporation
White Plains, New York

COBOL

Charles A. Phillips
Director, Data Systems Research Staff
Office Secretary of Defense

Wednesday, November 2

THEME: APPLICATIONS

9:00 a.m. Automation in the Field Army Supply System

Harold Silverstein
Assistant to the Chief Signal Officer (Operations Research and Automatic Data Processing Systems)
Department of Defense

10:30 a.m. Contributions of a Service Center to Data Processing

Jack Moshman
Vice President
C-E-I-R

1:30 p.m. Field Visits

1. C-E-I-R
   1200 Jefferson-Davis Highway
   Arlington, Virginia

2. Intelligent Machines Research Corporation
   7019 Edsal Road
   Alexandria, Virginia

3. R.C.A. Data Processing Center
   1725 K Street, N.W.
   Washington, D.C.

Thursday, November 3

THEME: APPLICATIONS

9:00 a.m. — Current Developments in Information Storage and Retrieval

Calvin Mooers, Proprietor
Zator Corporation
Cambridge, Massachusetts

10:30 a.m. — Case Example of Integrated Data Processing

William Hollis
Pittsburgh Plate Glass Company
Pittsburgh, Pennsylvania

1:30 p.m. — State Government ADP - A Report on Methodology

Edward F. R. Hearle
and
Raymond Mason
RAND Corporation
Santa Monica, California

3:00 p.m. — MICR and the Bank of America

Warren Prince
Manager, Business Data Processing
General Electric Company
Phoenix, Arizona

Friday, November 4

THEME: MANAGEMENT OUTLOOK

9:00 a.m. — Automation in Medical Science and Practice

Milton I. Schwalbe, M.D.
Medical Officer
Data Processing Staff
Veterans Administration

10:30 a.m. — Man - Machine Systems

Alex Orden
Director, Computing Center
University of Chicago

ADVISORY COMMITTEE FOR THE INSTITUTE

ALEXANDER, Samuel N. - Chief, Data Processing Systems Division, National Bureau of Standards

ANGEL, Herbert E. - Director of Administration, General Services Administration

BAILY, Nathan A. - Dean, School of Business Administration, The American University

BLANCHE, Ernest E. - President, Ernest E. Blanche & Associates, Inc.

BUSH, George P. - Emeritus Professor of Government and Public Administration, The American University

CUNNINGHAM, Joseph F. - Associate Director of Statistical Services, Comptroller of the Air Force

CURRY, Robert B. - Comptroller, Southern Railway System

DILLON, John H. - Administrative Assistant to the Secretary of the Navy

GLASER, Ezra - Statistician, National Analysts, Inc.

HARKIN, Duncan C. - Chairman, Department of Mathematics and Statistics, The American University.

HATTERY, Lowell H. - Professor of Government and Public Administration, The American University

MAHONEY, Edward J. - Assistant Director, Accounting and Auditing Policies Staff, General Accounting Office

McPHERSON, James L. - Machine Development Officer, Bureau of the Census

PHILLIPS, Charles A. - Director, Data Systems Research Staff, Office of the Secretary of Defense

PROVAN, John R. - Assistant Administrator for Administration, Veterans Administration

ROBINSON, Herbert S. - President, CEIR

SECKLER-HUDSON, Catheryn - Dean, School of Government and Public Administration, The American University

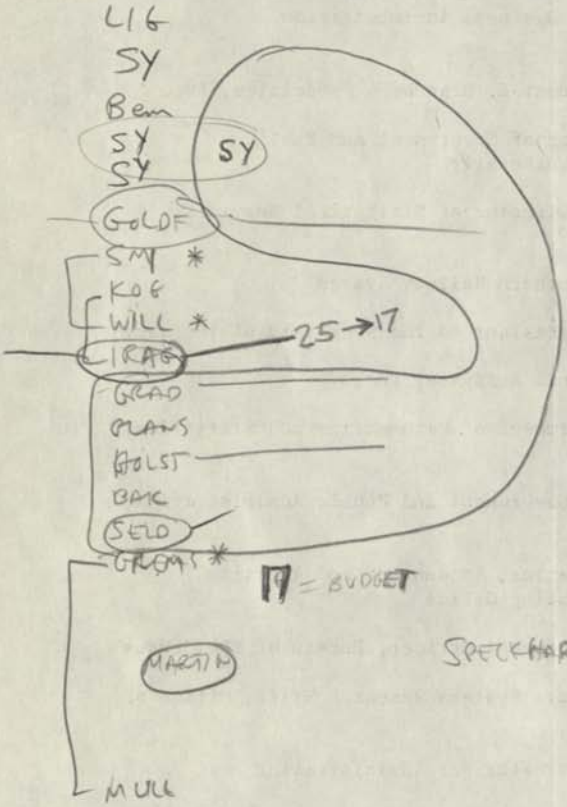
STICKNEY, George - Technical Assistant to the Fiscal Assistant Secretary (Systems and Methods), U. S. Treasury Department

RELATED INSTITUTES SCHEDULED FOR 1960-61 ARE:

Third Institute on Information Storage and Retrieval February 13-17, 1961	Sixth Institute on Research Administration April 24-28, 1961	Eighth Institute on Records Management May 15-26, 1961

HATTERY
McDONALD
PHILLIPS – COBOL FALL APART – SPLIT – PATCH NOW / NEXT MAY
DILLON CALLED, PASADENA, – WOULD HAVE EXEC C. MTG.
+ 2 CAMPS? TO 2 DEC

WOULD SEE ME IN N.Y. — NO, HARMON – RESPONS.



THE CHESAPEAKE AND OHIO RAILWAY COMPANY
TERMINAL TOWER · CLEVELAND 1, OHIO

October 28, 1960

Mr. Robert W. Bemer
Manager, Corporate Logical Systems Standards
International Business Machines Corporation
112 East Post Road
White Plains, New York

Dear Mr. Bemer:

I am very happy to learn that you have accepted an invitation to participate in our conference on "Business Languages". The fine work that you and IBM have done in this field is justifiably well-known, and we are sure that your contribution to our program will be very worthwhile.

In selecting the speakers for this meeting, we have concentrated on those in the field who are actively engaged in practical programming research. Our object in so doing is to present a user-oriented program. We feel that there is currently great interest among users on such questions as:

1. What are the latest developments in the construction of Business Languages and the application of these languages to computers?

2. How closely will we approach the goal of a common language for different computers and when will the goal be achieved?

3. What problems will the user encounter with these languages and what benefits will he gain?

The above is not, of course, a comprehensive list of the problems of interest in the area. We hope, however, that it will be of help to you in indicating the aims and approaches of the conference.

It is planned to distribute material to the registrants before the conference which will give them necessary background information and orient them to the talks and discussions. This technique has proved very valuable in the past in maximizing the rapport of the audience and promoting intelligent discussion. We would appreciate any contribution you could make by way of descriptive material, papers, charts, etc. A registration of approximately 300 is expected. We would also like a short biographical sketch. The biographical material will be reproduced and distributed by the Conference Committee.

At a later date we will mail you information relating to the detail arrangements for the conference. On behalf of the Cleveland-Akron Chapter of the ACM, I extend our gratitude for your offer to help in this program. I look forward to meeting you at the conference on June 15.

Very truly yours,

Warren F. Spalding

Warren F. Spalding
Conference Chairman

WFS:lc

cc: Mr. Jack Richardson
Mr. Ray Hitti

LOCKHEED AIRCRAFT CORPORATION
MISSILES and SPACE DIVISION · SUNNYVALE, CALIFORNIA

6 October 1960

Mr. Bob Bemer
Logical Systems Standards
IBM Corporation
112 East Post Road
White Plains, New York

Dear Bob:

On behalf of the San Francisco Bay Area Chapter of the Association for Computing Machinery, I would like to invite you to speak on the general subject of Problem Oriented Languages at our November 22nd dinner meeting. We are particulary interested in your comments on the status of IBM POL Projects and what you consider to be the basic elements of such languages. Please feel free to discuss any and all aspects of the very broad topic of Problem Oriented Languages.

The meeting has been scheduled for the Villa Chartier Restaurant on El Camino Real (Highway 101) in San Mateo. You might find it convenient to make your reservations either there at the Villa Hotel, operated in conjunction with the Villa Chartier or at the Hilton Inn at the San Francisco International Airport. I can highly recommend either place.

I've scheduled the social hour at 6 p.m., dinner at 7:30 — the business meeting (which usually lasts all of five minutes) will get started about 8:30, after which the floor is yours. Naturally, you are our guest for dinner.

If you can arrange to land earlier in the day, we'd be delighted to have you drop in for a visit. Lockheed Missiles is located in Sunnyvale on the Bayshore Freeway (Highway 101 Alternate or Bypass) immediately south of Moffett Field. Coming south on the Freeway, continue past the Moffett Blvd. overpass and Moffett Field South Gate (stop light). The next overpass is Mountain View-Alviso Road. Take the overpass! It is possible to take a left just as you get off the overpass — don't do it! Take the next left about half a block further down — then, another left at the Building 101 sign, and park on your immediate left in Visitor's Parking.

The guard will direct you to the Building 102 lobby. Have the receptionist call me, Lee Amaya, B. C.(Bernie)Dove, or any one else you can think of. It would be a pleasure to show you our set-up, especially if our 729IV's happen to be working.

Sincerely,

LOCKHEED AIRCRAFT CORPORATION
MISSILES AND SPACE DIVISION

Herb Finnie

C. H. Finnie, Jr. Head
Applied Digital Systems Development
Dept. 59-14 - Bldg. 102B
Plant 1 - Sunnyvale
REgent 9-4321 X28344

HF:sv

cc: T. W. Wilder

October 11, 1960

Mr. Joseph C. Batz
Program Chairman
General Electric Company
Court Street
Syracuse, New York

Dear Mr. Batz:

The title for my talk will be "International Standards and The Computer Field". I shall review the status of the national and international efforts for standardization in data processing as it affects the designer and user. The structure of the presentation will parallel closely the scope of the X-3 Committee of the American Standards Association. Specific areas are:

1. Elements of language
2. Data format
3. Common machine independent language
4. Operating systems

I shall attempt to put both the ASA and professional efforts in their proper relationship.

I expect the talk will run about an hour and would be very pleased if it were informal and produced many questions. I assume this is an evening meeting but do not have the time yet. I shall arrive some time in the early afternoon and hope it is possible to return on a 10:30 plane.

Cordially yours,

R. W. Bemer
Mgr, Corporate Logical
Systems Standards

RWB/ep

GENERAL ELECTRIC COMPANY

COURT STREET, SYRACUSE, NEW YORK . . . TELEPHONE GRanite 6-4411
BLDG. 9 - ROOM 52
COURT ST. PLANT
SYRACUSE, NEW YORK

DEFENSE ELECTRONICS DIVISION

HEAVY MILITARY ELECTRONICS DEPARTMENT

October 5, 1960

Robert W. Bemer
IBM Corporation
White Plains, New York

Dear Mr. Bemer,

We are looking forward to our meeting with you on Wed October 26th. Along with your prepared talk we wish to discuss with you our plans and activities for the 1962 ACM Conference to be held here in Syracuse.

I would be very grateful to receive a brief abstraction or outline of the topics to be covered in your talk as soon as possible in order to include it in the meeting notice to our members.

Sincerely yours,

Joseph C. Batz

Joseph C. Batz
Program Chairman
Syracuse Chapter - ACM

JCB/ahb

CC. D. G. Lewis
Ann Wang
R. S. Jones

re "INTERNATIONAL STANDARDS AND THE COMPUTER FIELD"

AA 147 (6) 1240 LGA - M 152 SYR
AA 573 (CV) 130 LGA - ALB - 328 SYR
AA 531 (CV) 330 LGA - 452 SYR (C-32, OCT)
AA 232 (6) 1030 PM ~~SYR~~ - 1231 LGA

CHUCK BRUCKEN
TO IBM OFFICE — LIMO
1000 JAMES ST

DATA PROCESSING DIVISION ELECTRIC TYPEWRITER DIVISION

IBM UNITED KINGDOM LIMITED

101 WIGMORE STREET · LONDON · W·1

Telegrams: INBUSMACH, WESDO, LONDON
Cables: INBUSMACH, LONDON
Telephone: WELBECK 6600

4th October, 1960.

Mr. R. W. Bemer,
IBM Corporation,
112 East Post Road,
White Plains,
New York.

Dear Bob,

I want to take this opportunity to express my appreciation for your considerable help to this Company by your address to the British Computer Society on Thursday last. All comments we have heard have been extremely favourable.

Mr. Gearing has asked me to ask you to make sure that when the papers are duplicated, which he tells me you promised to do for him prior to publication, the following legend should be included at the head of the paper. Gearing suggests that this should be done to protect all the parties concerned.

> "This paper formed the basis of a talk given by Mr. R. W. Bemer to the British Computer Society on the 29th September, 1960. It is not to be reproduced without prior permission of the author."

I enjoyed meeting you again and look forward to our next get-to-gether.

With best wishes,

Yours sincerely,

Nick

N. Jonas.

cc: Mr. D. J. N. Stirton

NJ/el

THE AMERICAN UNIVERSITY
SCHOOL OF GOVERNMENT AND PUBLIC ADMINISTRATION
WASHINGTON, D. C.

29th August, 1960.

R.W. Bemer,
IBM Corporation,
White Plains, New York.

CORPORATE LOGICAL
SYSTEMS STANDARDS

1960 AUG 31 PM 3:42

Dear Mr. Bemer:

This is to follow up our conversation at Endicott a few weeks ago concerning your participation in our Seventh Annual Institute on Electronics in Management.

We expect about 100 people, most of them from management positions in government and industry. We assume something more than average lay information about digital computers but not sophisticated knowledge about computers or programming. We believe that most are sufficiently informed for a profitable presentation and discussion about major developments in programming.

Because of your work in computer programming and your leadership in the American Standards Association attention to common languages, we hope you can present the subject of "Common Languages for Computers" at our Institute. The scheduled time is 1:00 p.m., Tuesday, November 2nd at 1901 F Street, N.W. I suggest a presentation of about 45 minutes. We will then ask Charles Phillips to make a brief statement about COBOL, to be followed by a question and answer period.

For the last hour in the afternoon we will break into smaller groups for discussion of special interests. I hope it may be possible to stay on through the afternoon. The detailed schedule will be:

1:00 - 1:45	"Common Languages for Computers"
1:45 - 2:00	COBOL
2:00 - 2:15	Recess
2:15 - 3:00	General discussion
3:00 - 3:15	Intermission
3:15 - 4:30	Group discussion

I look forward to a profitable session and hope very much you will be able to participate.

Sincerely yours,

Lowell Hattery

Dr. Lowell H. Hattery

GENERAL ELECTRIC COMPANY

COURT STREET, SYRACUSE, NEW YORK . . . TELEPHONE GRanite 6-4411

Room 52 - Building 9
Court Street Plant

DEFENSE ELECTRONICS DIVISION

HEAVY MILITARY ELECTRONICS DEPARTMENT

August 18, 1960

INFO - B106
STDS - ABSTRACT
APPLIC? PDL?

Robert W. Bemer
International Business Machines Corporation
White Plains, New York

Dear Mr. Bemer:

We of the Syracuse Chapter of ACM would be most grateful to have you speak to our members at one of our chapter meetings this year. We are looking forward to a close relationship with you as our New York State Regional Representative and with National ACM in general.

The topic could be of your own choosing but to aid you in making a selection, I will give you the following information about our membership. Most of the members are computer programmers, working both in scientific and business applications. Others are salesmen and sales support people from various manufacturers, managers of data processing groups and installations, and a few computer designers.

It would be preferable to have you as our guest at our second meeting of the season on Wednesday, October 26, 1960; however, if this is not convenient, we would be most happy to have you come at some other time. (We usually meet on the last Wednesday of every month.)

I will be attending the National ACM conference in Milwaukee next week and would like to discuss the matter with you if you will be there. Or, I may be reached at the above address or at GRanite 6-4411 ext. 8765.

Thank you sincerely for your gracious consideration.

Very truly yours,

SOFTWT ON VAC

Joseph C. Batz
Program Chairman
Syracuse Chapter, ACM

JCB/cam

cc: D.G. Lewis
Ann Wang

21 July 1960

Mr. N. Jonas
Data Processing Division
IBM United Kingdom Limited
101 Wigmore Street
London W1, England

Dear Nick:

Having been officially invited, I officially accept. I am giving a talk today to a group of University of Computer Center Directors and I particularly like the title. Perhaps you would have no objection if I use it for the talk before the Society. It is "Survey of Modern Programming Techniques". Could you make me an estimate of how long a period of boredom they will stand. The duration of the talk should also be a function of the time of day. Assuming this is in the evening, I need your guidance for a time limit.

I have made no definite plans yet but the peculiar timing of these two meetings suggest to me that I might be able to accomplish some needed liaison in the earlier part of the week. Subject to the approval of IBM UK and Bill Andrus, I might talk to some non-IBM personnel, perhaps in company of Messrs. Eadie and Dudeney. Such persons might include Mr. Tootill (on Glossary and Terminology) and Mr. Ross of Ferranti (on further code developments). I believe the air can be cleared for international standards if we continue to get together privately beforehand. Please let me know what you think of these suggestions. You may have some other people with whom a talk might be fruitful in your estimation. Perhaps a visit to the laboratory would benefit me.

I certainly wish to have three or four hours to talk with you to discuss the intricate politico-technical history of COBOL and CODASYL.

Cordially,

R. W. Bemer
Manager, Corporate
Logical Systems Standards

RWB/jes
cc: Mr. W. E. Andrus
Mr. I. C. Liggett

DATA PROCESSING DIVISION ELECTRIC TYPEWRITER DIVISION

IBM UNITED KINGDOM LIMITED

101 WIGMORE STREET · LONDON · W·1

Telegrams: INBUSMACH, WESDO, LONDON
Cables: INBUSMACH, LONDON
Telephone: WELBECK 6600

13th July, 1960.

Mr. R. W. Bemer,
IBM Corporation,
590 Madison Avenue
New York 22, N.Y.

Dear Bob,

British Computer Society

Further to the correspondence on the above, you will by now have heard that a meeting of the manufacturers committee has been called for Friday, September 30th. You have now been officially invited to address the Annual General Meeting of the above Society on the 29th September, and we all hope you will be able to make it.

I would be grateful if you could confirm that the necessary official permission has been given and perhaps you could let me know fairly soon how you would like your talk to be titled. We have to know this as an announcement will be made fairly soon. How about something fairly loose such as "Automatic Programming Techniques - Past, Present and Future"? However, I do leave this entirely up to you.

Looking forward to hearing from you and indeed seeing you in September.

Best wishes,

Yours sincerely,

N. Jonas.

cc: Mr. W. E. Andrus

June 8, 1960

IBM CONFIDENTIAL

TO: MR. I. C. LIGGETT

SUBJECT: British Computer Society

Mr. N. Jonas of IBM United Kingdom has corresponded with Mr. R. W. Bemer relative to Mr. Bemer speaking before the British Computer Society on his next trip to England. The proposed subject is "Automatic Programming Development in the United States".

There has been no date as yet set for this meeting since it is planned to coordinate a meeting of the British Computer Society with the next meeting of the ICT Automatic Programming Committee. As you know, the ICT Committee is the group to which Mr. Bemer spoke on February 29th of this year. At their next meeting, they will again discuss the subject of character sets and it is felt that Mr. Bemer's attendance would be most appropriate in view of the pending activity on this subject.

I believe it appropriate for Mr. Bemer to attend the ICT meeting as well as to speak before the British Computer Society. I will indicate to you the meeting dates as soon as they have been established.

W. E. Andrus, Jr.

WEA:mcm
cc: Messrs. R. W. Bemer
N. Jonas

DATA PROCESSING DIVISION ELECTRIC TYPEWRITER DIVISION

IBM UNITED KINGDOM LIMITED

101 WIGMORE STREET · LONDON · W·1

Telegrams: INBUSMACH, WESDO, LONDON
Cables: INBUSMACH, LONDON
Telephone: WELBECK 5600

26th April, 1960.

Mr. R. Bemer,
IBM Corporation,
112 East Post Road,
White Plains,
New York.

Dear Bob,

We have had a request from the British Computer Society asking us to see whether you personally would be prepared to give a review paper on the Development of "Automatic Programming Routines for Business in the USA" at a meeting which will be specially arranged to co-incide with the next trip to London which I presume will be for a Committee Meeting of the ICT Committee.

I am just writing unofficially to obtain your feelings on this subject and to state that we would be very grateful if you could undertake this as it would without doubt help IBM UK in their general sales effort. Perhaps you would let me know if you approve or disapprove of this suggestion, and, if you are willing to go ahead, whether I should make some official representation through any other persons.

I am looking forward to meeting you again when you are next in the UK.

Yours sincerely,

N. Jonas.

NJ:el

session II | logic design

Thursday, 28 July, 2:00 p.m.

Chairman — Howard E. Tompkins, Professor of Electrical Engineering, University of New Mexico, Albuquerque, New Mexico.

THE LOGICAL STRUCTURE OF THE MANIAC III ARITHMETIC UNIT — David Jacobsohn, The University of Chicago, Institute for Computer Research, Chicago, Illinois.

A STRAIGHTFORWARD WAY OF GENERATING ALL BOOLEAN FUNCTION OF N VARIABLES USING A SINGLE MAGNETIC CIRCUIT — K. V. Mina and E. E. Newhall, Bell Telephone Laboratories, Murray Hill, N. J.

Coffee Break —

TUNNEL DIODE LOGIC AND MEMORY CIRCUITS — W. F. Chow, Electronics Laboratory, General Electric Co., Syracuse, N. Y.

DESIGNING MULTIPLE-OUTPUT SWITCHING CIRCUITS — John Earle and G. A. Maley, Product Development Laboratory, Data Systems Division, International Business Machines Corp., Poughkeepsie, N. Y.

TWO-DIMENSIONAL PARITY CHECKING — Peter Calingaert, The Computation Laboratory of Harvard University, Cambridge, Mass.

session III | systems

Friday, 29 July, 9:00 a.m.

Chairman — David C. Evans, Chief Engineer, Bendix Computer, Division of the Bendix Corporation, Los Angeles, California.

INPUT-OUTPUT CONTROL ON A KILOMEGACYCLE COMPUTER — Charles Blair, Department of Defense, Washington, D. C.

A DIGITAL FILTER — Walter C. Davis, BJ Electronics, Borg-Warner Corporation, Santa Ana, California.

Coffee Break —

A METHOD FOR PERFORMING CORRECTIVE MAINTENANCE ON A DIGITAL COMPUTER BY ROTE PROCEDURE — H. Bullard, Great Valley Laboratory, Burroughs Corporation, Paoli, Pa.

PROGRAMMING SYSTEMS AND COMPUTER DESIGN — Walter Jacobs, Department of Defense, Washington, D. C.

SYSTEM ORGANIZATION OF FLIP (Floating Indexed Point arithmetic) Stanley Hanson, Argonne National Laboratory, Lemont, Illinois.

session IV | seminars

Three parallel seminar sessions will be held to permit informal discussion of the papers with the authors and to hear contributions from other conferees working in the same field.

2:00 p.m. Casino
LOGIC DESIGN
Howard E. Tompkins

2:00 p.m. Manor East
COMPONENTS AND CIRCUITS
Tudor Finch

2:00 p.m. Manor West
SYSTEMS
David C. Evans

BENDIX

7th Annual Symposium on
Computers and Data Processing

28-29 July 1960

The Stanley Hotel Estes Park, Colorado

Sponsored by the Electronics Division

DENVER RESEARCH INSTITUTE
UNIVERSITY OF DENVER

general information

about the symposium

An air of informality prevails at this symposium. A "bull session" room has been set aside where conferees can meet old friends and make new ones. There will be no exhibits, no recruiting posters, and no proceedings.

location and transportation

The Sympsium will be held at the Stanley Hotel, Estes Park, Colorado. Estes Park is clearly shown on highway maps. 70 miles northwest of Denver on good paved highways. Driving time from downtown Denver or the airport is about 2½ hours. The only public transportation is by Rocky Mountain Parks Transportation Company from the Greyhound Bus Depot, 17th and Glenarm, on the following schedule:

Leave Denver	Arrive Stanley	Leave Stanley	Arrive Denver
9:00 a.m.	12:05 p.m.	8:30 a.m.	12:05 p.m.
2:30 p.m.	6:00 p.m.	12:45 p.m.	3:10 p.m.
7:00 p.m.	9:00 p.m.	3:15 p.m.	6:25 p.m.

The Stanley Hotel is a regular stop and all buses also stop at the railway station in Denver 15 minutes later than the schedule shown. Fare is $5.50 one-way, $8.80 round-trip. An additional 30 minutes should be allowed from downtown Denver to the airport. In addition to the above schedule, a special bus will leave the Stanley at 5:00 p.m. on Friday, arriving in Denver at 7:30 p.m.

All major car rental agencies are represented in Denver and at the Denver Airport, and Hertz also has an agency in Estes Park. Advance reservation is strongly recommended.

For assistance in transportation or other arrangements after arriving in Denver, conferees may call the University of Denver switchboard, SHerman 4-1811.

accommodations

The Stanley Hotel operates on the American Plan and has given a special rate to conferees and their families of $17.00 per day per person and $13.00 per day per child under the age of 12, exclusive of tax and tip.

The hotel reduces charges proportionately for meals not taken on days of arrival and departure. To obtain the special conference rate, please use the enclosed Hotel Reservation Form.

Because the Stanley is a resort hotel, very few single rooms are available, and it will be necessary under the price arrangements for conferees to room together in double rooms with twin beds. If conferees desire other accommodations, it is with the understanding that the price will be higher. All buildings at the Stanley are heated, have been recently redecorated, and are far above average in comfort. The Stanley will confirm all reservations in writing to conferees, after receipt of the original of the enclosed hotel reservation form. Please list DEFINITE ARRIVAL AND DEPARTURE DATES. Arrangements for extended stays at the hotel must be made with the hotel.

The Stanley dining room is open from 7:30 till 9:00 a.m., from 12:30 till 2:00 p.m. and from 6:30 till 8:00 p.m. A continental breakfast is served in the Lariat Lounge after 9:00 a.m.

registration

Advance registration should be made on the enclosed form and returned with a check for $25.00 to cover the registration fee. CHECKS SHOULD BE MADE PAYABLE TO THE UNIVERSITY OF DENVER. The conference check-in desk at the Stanley Hotel will be open beginning at 1:00 p.m. Wednesday, July 27th.

clothing

Informal attire is recommended. Temperatures average in the mid-70's and low 80's in the daytime and summer clothes are comfortable. Evenings are usually chilly by comparison, with the temperature as low as 50 degrees. A warm jacket is recommended, particularly for those who plan even short drives into the higher elevations in Rocky Mountain National Park.

activities — recreation

Conferees and their families will have the use of all hotel facilities including a heated swimming pool, tennis, golf, and dancing. Horseback riding and mountain tours are available at additional cost. Fishing spots abound nearby. Short mountain hikes start directly from the hotel, and it is only a short drive to a variety of high mountain scenery. The Stanley has a full-time Social Director and a junior hostess who takes children for the entire day including mealtimes. Baby sitters are available in the evenings at extra cost. Mountain bus tours leave daily from the Stanley Hotel.

7TH ANNUAL SYMPOSIUM ON COMPUTERS AND DATA PROCESSING

28-29 July 1960

session I — components and circuits

Thursday, 28 July, 9:00 a.m.

9:00 a.m.—Opening Remarks—Shirley A. Johnson, Jr., Director, Denver Research Institute.
Chairman—Tudor Finch, Special Systems Development Engineer, Bell Telephone Laboratories, Murray Hill, N. J.

ONE AMPERE TEN MILLIMICROSECOND COMPUTER DRIVING TRANSISTOR — C. H. Knowles, Semiconductor Products Division, Motorola, Inc., Phoenix, Arizona.

CHARACTERISTICS OF THIN-FILM PARAMETRONS — E. P. Stabler, Electronics Laboratory, General Electric Co., Syracuse, N. Y.

Coffee Break —

CIRCUITRY OF FLIP (Floating Indexed Point arithmetic) — Forrest Salter, Argonne National Laboratory, Lemont, Illinois.

THE MICROWAVE PHASE LOCKED OSCILLATOR AS AN ULTRA HIGH SPEED COMPUTER ELEMENT — G. B. Herzog, Radio Corporation of America, Princeton, N. J.

THE TWISTOR — PAST, PRESENT, AND FUTURE — Andrew H. Bobeck, Bell Telephone Laboratories, Murray Hill, N. J.

Association for Computing Machinery

Dallas-Fort Worth Chapter

DINNER MEETING

Tuesday, July 26, 1960
Western Hills Motel
Euless, Texas

Cocktails - 6:00 p.m.
Dinner - 7:00 p.m.
$3.75

A BUFFET DINNER WILL BE SERVED

Speaker - Robert W. Bemer, Manager, Logical Systems Standards, IBM

Subject - "ALGOL 60 and Its Implications"

Printed material will be available for you at the meeting.

BE SURE AND READ THE NOTICE FOR THE AFTERNOON SESSION.

A door prize will be given to paid-up attending members.

Contact your company representative or mail a check for $3.75 to

J. B. Harvill, Roselawn Drive, Denton, Texas

Association for Computing Machinery

Dallas-Fort Worth Chapter

TECHNICAL MEETING

Tuesday, July 26, 1960
IBM Building, 2911 Cedar Springs Road 2:00 - 4:00 p.m.
Dallas, Texas

Robert W. Bemer, Manager, Logical Systems Standards for IBM, will conduct a technical meeting. Mr. Bemer will discuss and answer questions concerning ALGOL (International Algorithmic Language).

This is an excellent opportunity to learn, in greater detail, about ALGOL and to ask Mr. Bemer any questions you may have.

INFO (COMM. ACM, NY REG)

"ALGOL 60 & ITS IMPLICATIONS"

JACTON - SALE
MACRO-DIR

WHAT IS ALGOL? - BRIEF HISTORY, SCIENTIFIC WITH POWER

NEG FACTIONS

DON'T WORRY NOW
MANY CHANGES, EVEN IN PSB, 60 (ALL OK)

STATUS — GERMANY
US — IBM
BURROUGHS (RESTRICTED ALGOL 58)
N. CAR 1105
U CHI,
MICH 704 - MAD
1604 - MELLAC
SDC - CLIP
CARNEGIE TECH - CORC GATE
RAND - AP2
ALGOL IMPLEMENTERS GROUP - PERLIS.

CHANGES - P. 20. DECL.

OPTIM. CODING

6 TO FAM, THEN G-20

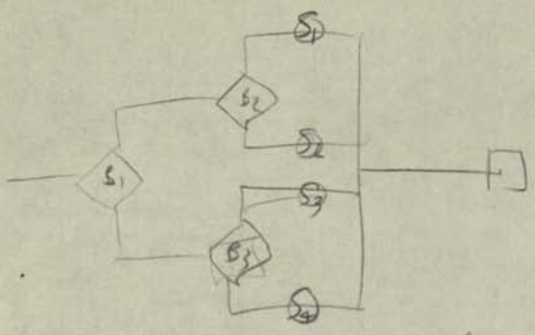
WRITE IN OWN 25/1

STDS - NOT YET READY, ANALOGY TO COBOL (EVEN WORSE)
SHORTCOMINGS - NO I/O OR SYST. PROCEDURE ONLY, INSUFF. FLOW SYNTAX, LOUSY CONDITIONALS (IF THEN BRACKETS)
PROBS ADDITIONS - SYMB. MANIP - SYMBOL MANAGER
JOE W.'s SECTION IN COMMUN IS OUTMODED, MUST BE REVISED LATER. PROBABLY

INTL STDS - BLEND ALGOL/COBOL - BACKUS META-SYNTAX.
ISO - ASA - X3, BSI

IMPLICATIONS 1. EQUIPMENT DESIGN
CHAR SETS
LIST STRUCTURE MACHINES, TABLES, INTERP (ATLAS - GILL)

2. USAGE - C
EDUCATE COMPUTER, INTERCHANGE, INTL.
PROMOTE CENTRAL COMP, STRETCH STYLE, COMMUNICATIONS
INCREASED POWER & FLEXIBILITY, RECURSION.



IF B_1 THEN { IF B_2 THEN S_1 ELSE S_2 } ELSE

IF STMT

IF CLAUSE

CS

Customer Executive Program . Endicott, N. Y.

DIRECTORS OF UNIVERSITY COMPUTING CENTERS
Class No. 6090
July 18 to 22, 1960

instructors

J. A. Kearns,
Course Manager
J. D. Hosie,
Coordinator

Adams, John F., Assistant Vice President Temple University	Philadelphia, Pa.
Andrews, Fred C., Associate Professor of Mathematics University of Oregon Graduate School	Eugene, Ore.
Belsky, Lawrence E., Ass't. Director, Statistical & Research Services Boston University	Boston, Mass.
Carter, Richard I., Director, Computation Center Northeastern University	Boston, Mass.
Corbato, Fernando J., Associate Director Massachusetts Institute of Technology	Cambridge, Mass.
Davis, Fred, Director, Computing Center Mississippi State University	State College, Miss.
DesJardins, Robert B., Manager of Computer Center Florida State University	Tallahassee, Fla.
Dolch, John P., Director - Computing Center State University of Iowa	Iowa City, Iowa
Duffin, John H., Professor of Chemical Engineering San Jose State College	San Jose, Calif.
Fender, Frederick G., Director, Computation Center Rutgers University	New Brunswick, N. J.
Fields, Raymond I., Assoc. Prof. of Math. & Director of Computing Lab. University of Louisville	Louisville, Ky.
Freiberger, Walter F., Associate Professor of Applied Mathematics Brown University	Providence, R. I.
Freund, Rudolf, J., Assoc. Prof. & Director Computing Center Virginia Polytechnic Institute	Blacksburg, Va.
Gallie, Thomas M., (Jr.), Ass't. Professor of Mathematics Duke University	Durham, N. C.
Garber, Morris J., Assistant Biometrician University of California	Riverside, Calif.
Gennaro, Joseph, J. Stevens Institute of Technology	Hoboken, N. J.
Gregory, Robert, Assistant Director, Computing Center University of Texas	Austin, Tex.
Hart, John F., Secretary, Computer Committee University of Western Ontario	London, Ont.
Hestenes, Magnus R., Full Projessor University of California	Los Angeles, Calif.
Hoffman, Walter, Director, Computing Center Wayne State University	Detroit, Mich.
Hollingsworth, Jack, Director Computer Laboratory Rensselaer Polytechnic Institute	Troy, N. Y.

	First Printing
[illegible]oodes, Robert A., Ass't Managing Director Littauer Statistical Laboratory	Cambridge, Mass.
Howell, Floyd (Jr.), Director of the Computer Center University of Miami	Coral Gables, Fla.
Kain, Frederick T. (Jr.), Assistant Comptroller Temple University	Philadelphia, Pa.
Keenan, Thomas A., Director, University Computing Center University of Rochester	Rochester, N. Y.
Laird, Donald T., In Charge, Computation Center Pennsylvania State University	University Park, Pa.
Macon, Nathaniel, Dir. of Computer Lab. & Prof. of Mathematics Auburn University	Auburn, Ala.
Maginnis, James B., Director, Computing Center Drexel Institute of Technology	Philadelphia, Pa.
Myers, Basil R., Prof. of Electrical Engineering & Chairman of the Dept. The University of Waterloo	Waterloo, Ont.
Nelson, Richard W., Research Associate, Physics & Math. Institute of Paper Chemistry	Appleton, Wisc.
Newhouse, Albert, Ass't. Director, Computing Center, Prof. of Math. University of Houston	Houston, Tex.
Prager, William, Computing Center Director Brown University	Providence, R. I.
[illegible]eeves, Roy F., Director, Numerical Computation Laboratory Ohio State University	Columbus, O.
Remmenga, Elmer E., Chief, Computing Center Colorado State University	Fort Collins, Colo.
Rheinboldt, Werner C., Director, Computing Center Syracuse University	Syracuse, N. Y.
Rolf, Howard L., Director of Computer Center Vanderbilt University Computer Center	Nashville, Tenn.
Rymer, Harry R., Director of Computing Center Northwestern University	Evanston, Ill.
Scroggs, James, Director, University Computer Center University of Arkansas	Fayetteville, Ark.
Selfridge, Ralph G., Director, Computing Center Miami University	Oxford, O.
Sherman, Gordon R., Ass't. Director Computing Center The University of Tennessee	Knoxville, Tenn.
Shull, Harrison, Director, Research Computing Center Indiana University Research Computing Center	Bloomington, Ind.
Smith, Howard M. (Jr.), Director of Computing Center University of Vermont	Burlington, Vt.
Snyder, James N., Director - 650 & Illiac Computing Center University of Illinois	Urbana, Ill.
[illegible]arpey, Paul, Data Processing Chief University of Kentucky-Computing Center	Lexington, Ky.
Townsend, Bill B., Director of Computing Center Louisiana State University Research Center	Baton Rouge, La.
Viavant, William, Director of Scientific Computations University of Oklahoma	Norman, Okla.

Guest Speakers

William F. Atchison Georgia Institute of Technology - Rich Computer Center	Atlanta, Ga.
M. P. Barnett Massachusetts Institute of Technology	Cambridge, Mass.
Robert C. F. Bartels, University of Michigan, Computing Center	Ann Arbor, Mich.
Elliott L. Buell Worcester Polytechnic Institute	Worcester, Mass.
Almand R. Coleman University of Virginia	Charlottesville, Va.
Richard H. Hill UCLA	Los Angeles, Calif.
J. R. Hackson UCLA	Los Angeles, Calif.
William B. Kehl University of Pittsburgh	Pittsburgh, Pa.
Sydney MacD. Lamb University of California	Berkeley, Calif.
Philip M. Morse Massachusetts Institute of Technology	Cambridge, Mass.
Elliot I. Organick University of Michigan	Ann Arbor, Mich.
Albert N. Schrieber University of Washington	Seattle, Wash.
Robert L. Smith, Jr. Texas A&M	College Station, Tex.
A. Wayne Wymore University of Arizona	Tucson, Ariz.
David M. Young, Jr. University of Texas	Austin, Tex.

IBM Guest Speakers

R. W. Bemer Applied Science Division	White Plains, N. Y.
F. Brooks Research Department - Mohansic Laboratory	Yorktown Heights, N.Y.
C. C. Hurd Industrial Process Control Laboratory	Peekskill, N. Y.
D. V. Newton Applied Science Division	White Plains, N. Y.
[illegible]. B. Poland Product and Development Laboratory	Poughkeepsie, N. Y.

Minutes of Fourth DSD Evaluation Board Meeting
Thursday, June 2, 1960

At the meeting the following members of the Board were present: Dr. DeCarlo, Chairman, Mr. Clark, Mr. Femmer, Dr. Cadden (representing Mr. Marcy), and Dr. Thomsen, Secretary. Invited guests present were: Mr. Bemer, Mr. Harmon, and Mr. Goodhue.

Mr. Bemer made the feature presentation on programming and discussed new development areas such as symbol manipulation, current problem areas, and recommendations. Details are given in the attached summary made available to Board members and invited guests prior to the meeting.

At the meeting Mr. Bemer emphasized the following points:

(1) No common machine language is possible contrary to popular opinion - simple illustrations were given to illustrate the point.

(2) Programming costs for a customer are initially large as is generally realized but after the initial period of an installation those costs do not decrease as is often believed. (Typical costs of programming - The cost of programming a 704 to cost of rental, for example, is in a ratio of one to one according to Mr. Bemer - in a 7090 it is likely to be in a ratio of three to one.)

Decision: This second point came as a combination shock and surprise to several members of the Board. Mr. Clark accepted the point but stated that documentation would be most desirable, and the Board concurred that this should be obtained in at least two cases. Dr. DeCarlo suggested our own 704 computation lab at Poughkeepsie as one possibility.

(3) With the increase in machine speeds and the complexity of machine organization programming, efforts are lagging in spite of all the recent significant advances. Thus the inefficiency of several slow machines is actually more desirable to some customers than one much faster system.

The Board centered considerable debate and discussion on two major points:

(1) The implications of programming in relation to systems design.

The desirability of mixing programming and engineering talent in the design of a system was pointed out in a variety of ways. The following comments were pertinent:

(a) Dr. Cadden asserted there was a lack of programming competence in machine design -- engineers seem to stop at machine language.

(b) Mr. Femmer called attention to the fact that the engineer does not ordinarily regard a machine from the point of view of the user.

(c) Mr. Clark, however, commented he did not believe it possible to find one superhuman person who can look at the problem of design from both viewpoints.

(d) Dr. DeCarlo acknowledged that it is entirely possible for programming to be unable to cope with a radical development in "componentry." He cited as an example the possibility of a jump of several orders of magnitude in the capacity of a Read Only Memory.

Decision: In the course of this discussion it was decided that it would be beneficial for a limited number of key engineers under the project managers in the systems area to acquire as soon as possible substantial programming knowledge. Mr. Femmer offered and agreed to implement this policy at once. Harmon concurred and offered to give programming instruction in Poughkeepsie as soon as appropriate.

(2) The vulnerability of our "bridging" strategy.

Dr. DeCarlo pointed out that our machines with their associated programming systems have not made it too easy for a customer to switch abruptly to a competitor's system. It is not possible to make such a change so easily as in the case, for example, with an automobile -- the owner of one model can acquire

a competitor's model inside of hours or even minutes. There are definite signs, however, that this situation will not last -- most of our competitors can make FORTRAN available now in their product line. The problem is when are we going to become vulnerable to a breakdown in this "bridging" between "models" which programming systems help to provide.

Decision: Dr. DeCarlo asked that we have a report as to when we will be vulnerable to the possibility of the customer's ability to make a quick change of a system in the "automobile" sense. This report should be categorized by machine family and should show estimated dates of "vulnerability" for each family. Mr. Harmon agreed to contact the appropriate people in IBM and obtain the necessary information for such a report.

R. W. Bemer

PROGRAMMING

OUTLINE OF PRESENTATION

I) Background Information.

a. Short history of programming in IBM and elsewhere. Relative strength positions of U.S. competitors, foreign competitors, universities, etc.
b. The education of the computer - generalized programming versus applications. Comparable to the difference between academic course and technical training.
c. Types of programming systems - graded from machine language thru assembly programs, simple macro-instructions, macro-generators, compilers, interpreters, compiler-interpreters, procedure-oriented languages (FORTRAN, COMTRAN, COBOL), problem-oriented languages (sort, merge, report generators, file maintenance and updating), input-output control systems, operating systems, diagnostics.
d. The scientific vs. commercial split, scheduled to disappear in many respects.
e. Language development and standardization. FORTRAN leading into ALGOL thru ACM-GAMM. (How to convert by machine). CODASYL, aims and practice. UNCOL experiment of SHARE.

2) Areas of New Development.

a. Symbol manipulation, strings and list structures. How scans work.
b. Sub-languages (such as Backus' meta-language, Shapiro's tables) and super-languages (jargons, Barnett's SHADOW). The cascading principle in development.
c. What is UNCOL? What should a common sub-language be at the point where it starts to exhibit machine-dependent characteristics?
d. Making procedure language easier for the user. First the separation of data description, then time dependencies and flow (Lourie's GENESIS), then storage allocation, then environment and traffic. How far along are we now?
e. New languages of different form. TABSOL (2-dimensional), systems languages.

3) Present Problem Areas.

a. Too large a variety of machines. All old packages must be duplicated and improved for all new machines.
b. No basic tools are in common use to aid production of programming systems. This could be called auto^2-programming. Use of bootstrapping to create new systems.

c. Insufficient time between production and delivery for testing both programming systems and customer programs utilizing these systems. Time wasted by field calls, SHARE, GUIDE, etc. Simulation programs useful, but expensive.
d. Relative low speed of programming production compared to engineering advances and easier hardware production.
e. Interchange of programs, compatibility thru machine family, machine-independent languages, partial standards.
f. High cost of educating machines partly due to loss of knowledge, lack of dissemination of new techniques because of press of work (publications, how-to manual). Own tools not being used efficiently and to advantage thru macros, etc. (SOS).
g. Balance is too high in favor of machine line programming, rather than functional. Suitability of sorts, assemblies, mathematical subroutines, report generators, etc. to functional attack.
h. Difficult to obsolete and scrap old programs.

4) Needs.

a. More programmer education. (for non-programmers)
b. Automatic assembly program generator, simulator generator.
c. More functional alignment of programmers.
d. Better liaison between divisions, both programming and engineering, to achieve standards in hardware and I/O equipment.
e. A useful systems language, including standard operating system specs.
f. Manuals and texts for program production.
g. Algorithms for automatic storage allocation, traffic control, scheduling.
h. Standard hardware elements for all computers in the line.
Read-only memory (over 75% of instructions are static in present programs)
Real-time clocks Serial Reader
Indexable instruction counters
Standard data format (word marks are bad)
Larger character sets
Binary pattern output
Access to all status indicators, every traffic possibility so indicated.
i. A stronger sense of urgency and professionalness.

5) Recommendations.

a. More manpower to write working ALGOL Commercial language translators embedded in standard operating systems.
b. Stronger emphasis on tools, functional methods.
c. Rotating training of engineers in programming and vice versa.
d. Severe restrictions on variety of machines produced.

MEMORANDUM

FSD Headquarters
Rockville, Maryland
May 24, 1960

To: Mr. R. W. Bemer, CHQ

Subject: Your Participation in the Multiprocessing Conference - May 19, 1960

Bob, may I personally compliment you upon the excellent contribution you made to our Multiprocessing Conference and express to you the appreciation of the Federal Systems Division for your participation.

Your presentation displayed the greatest depth of thought concerning the impact of advanced multi-programming systems on machine organization. Your points were extremely relevant and well received.

Please accept our appreciation for a job well done.

D. L. Dittberner

DLD:moc

cc: Mr. J. B. Jackson
Mr. E. I Liggett, CHQ
Mr. J. C. McPherson, CHQ
Dr. E. R. Piore, CHQ

24 May 1960

Memo To: Mr. D. L. Dittberner,
FSD HQ, Rockville, Md.

Subject: Hardware Features for Simpler Programming

Per your request, here is the substance of the statements I made at the Multi-processing Conference on 19 May.

1. There is no such thing as percentage of compatibility for programs between different computers. There is only percentage of additional work required to modify the programs to work on a different machine.

2. Hardware developments are achieving greater magnitudes of increased speed than are possible by programming. Therefore, the hardware facilities must be rebalanced to favor programming.

3. It is impossible in any practical sense to translate between machine languages. Even in related families of machines there are tremendous difficulties. This is due to the introspective nature of machine language and the ability to modify programs by occurring data.

4. A distinction must be made between problem-oriented languages and procedure-oriented languages.

5. The UNCOL concept does not seem feasible to me, for it must embrace all possible existing and future characteristics of computers.

6. Multi-processing concepts add to programming difficulties under existing rules. I cannot imagine an instance where Multi-processing eases programming restrictions in any way that could not be done also in simple processing.

The following points relate to hardware design and control of program fabrication. Most of these are vital to successful Multi-processing even though their effect may be secondary, such as reducing storage requirements.

1. The computer should be designed so that the programmer may operate upon bit streams of arbitrary and variable length. Parallel byte mechanisms are permissible as a convenience. I suggest that all input-output equipment (eventually) be of a pure bit-pattern type similar to photo-engraving. This allows an infinite set of characters of any size.

2. All states of the computer and any interconnected devices must be accessible to the program through indicator triggers. These states are not necessarily binary.

3. There must be indexed instruction counters and related features such that the program may be assembled once to produce relative absolute coding (see Rex Rice on this subject). Under these rules the program could also be written directly in relative absolute.

4. With the facilities of index registers and indirect addressing, the number of instructions actually modified in present day programs is well under 10%. This points up a large advantage for a read-only memory of the total, not only from the reduced cost standpoint but from the increased organization in programming that accrues through the use of Execute instructions. In addition, reliability is increased since there is no need to rewrite and verify in memory after a destructive read-out. Parity bits may also be permanent in a read-only memory. Read-only memory forces an elimination of modified instructions which is good for compatibility.

5. The Execute instruction is vital for other modes of operation, particularly simulation. Devices like the existing Convert instruction are very useful.

6. The facility should exist to operate with the simpler instructions and primitives of micro-programming, but with the capability of assigning groups of such instructions to parallel execution. This might be done by a program-settable matrix. Here we seem to revert to the Turing machine concept, to synthesize at will when needed. This concept is not in conflict with the larger operations such as Convert, Floating Point Arithmetic, etc., which are common because they have a high pay-off.

7. Variety of both machine type and configurations of a single machine should be kept at a minimum. Modularity, yes -- but only of a type which is susceptible to automatic handling by program generation.

8. There must be standards for collating sequences and character representation, for these build the programming interface which corresponds to hardware interface.

9. The machine should be built with all possible recursive ability (see the G20 for a minor example). The heart of information processing is access to elements of related sets. List structures provide the optimum means of finding data, between sorting into ordered files on one hand and a complete search for a key on the other hand. Contrary to RAMAC organization, lists provide for multiple set membership without physical duplication in the storage media.

10. The most promising new area of programming development is string manipulation for operating upon symbols, names and values. To my mind this is our one avenue of creating universal language for a multitude of machines without duplicating the translating processor, for each machine at a correspondingly large cost. With string manipulation we may translate between the external identifiers and names used by humans into more compact internal identifiers suitable to machine processes.

11. Facilities should exist for on-line interpretation of source language. This is in contrast to compilation and is of value when only a small portion of the program combinations are used in actual practice. This is particularly valuable for military and real-time applications, since the size of the high speed store may be reduced.

R. W. Bemer
Manager, Logical
Systems Standards

RWB/jes

cc: Mr. J. E. Griffith, CHQ
Mr. I. C. Liggett, CHQ

FSD Headquarters
Rockville, Maryland
May 10, 1960

MEMORANDUM

TO: See Attached Distribution List

SUBJECT: Multiprocessing Conference; May 18th and 19th; CHQ

The concepts of multi-programming and multiple computer operation have long been of theoretical interest. The recent introduction of the RW 400, the Burroughs Modular Computer, the IBM France "STRIDA II" Computer, the GAMMA 60, and others, presents us with many new technical concepts of machine organization of practical concern.

Certain of these concepts appear to offer unique advantages to users in the military market.

You are invited to attend a conference, organized by FSD, which has as its purpose the review, discussion, and evaluation of these new concepts in Multiple Processing that are presently being marketed to the Federal Government.

The emphasis of this conference will be on the advantages which are being offered, how competition is achieving these, alternative methods of achieving these desired advantages; and, finally, the stimulation of future thought in finding an optimum means of providing these advantages to IBM Customers.

The conference will be held at Corporate Headquarters on the 17th floor. It will start Wednesday, May 18th at 10:00 AM, and is expected to close on Thursday, May 19th at 4:30 PM. Your active participation will be appreciated.

Please respond if you plan to attend, in order that adequate seating may be insured.

John B. Jackson

John B. Jackson,
Director of Planning

JBJ:ns

DISTRIBUTION LIST

CORPORATE RESEARCH

Mr. J. W. Backus - Mohansic
Miss E. M. Boehm - Mohansic
Mr. F. P. Brooks - Mohansic
Mr. J. Cocke - Mohansic
Mr. W. R. Elmendorf - Mohansic
Mr. J. E. Griffith - CHQ
Mr. H. Hellerman - Mohansic

CORPORATE

Mr. R. Bemer - CHQ
Mr. J. G. Fleming - CHQ
Mr. W. H. Johnson - CHQ
Mr. E. I. Liggett - CHQ
Mr. J. C. McPherson - CHQ
Mr. P. M. Rider - CHQ
Mr. H. D. Ross - CHQ
Mr. D. Sayre - CHQ

GENERAL PRODUCTS DIVISION

Mr. G. Benson - Endicott

DATA SYSTEMS DIVISION

Mr. J. T. Ahlin - White Plains
Mr. J. Blau - Poughkeepsie
Mr. W. Buckholtz - Poughkeepsie
Mr. T. Codd - Poughkeepsie
Mr. J. Greenstadt - CHQ
Mr. A. L. Harmon - White Plains
Mr. W. P. Heising - CHQ
Mr. B. Lappel - Poughkeepsie
Mr. S. Lida - Poughkeepsie
Mr. M. Paley - Poughkeepsie
Mr. D. Pendery - Poughkeepsie
Mr. P. Seaman - Poughkeepsie
Mr. C. Smith - Poughkeepsie
Mr. D. Sweeney - Poughkeepsie

ADVANCED SYSTEMS DEVELOPMENT DIVISION

Mr. B. Brown - Mohansic
Mr. P. Crawford - White Plains
Mr. D. V. Newton - White Plains
Mr. J. Orzano - White Plains
Mr. C. C. Smith - White Plains
Mr. H. T. Ware, Jr. - White Plains

FEDERAL SYSTEMS DIVISION

Mr. J. D. Aron - Wash.
Mr. J. Batley - Bethesda
Mr. D. L. Dittberner - FSD-HQ
Mr. N. P. Edwards - FSD-HQ
Mr. J. J. Fox - Wash.
Mr. T. A. Gorman - Wash.
Mr. A. W. Heineck - Kingston
Mr. C. Hesner - Bethesda
Dr. T. R. Horton - FSD-HQ
Mr. J. J. Kossuth - FSD-HQ
Mr. R. L. Rockefeller - Kingston
Mr. J. Selfridge - Kingston
Mr. W. B. Teague, Jr. - FSD-HQ
Mr. K. Van Mechelen - Owego
Mr. E. C. Weiland - Wash.

PRELIMINARY AGENDA

MULTIPROCESSING CONFERENCE

Time: May 18th, 10:00 AM to 4:30 PM
May 19th, 9:30 AM to 4:30 PM

Location: Corporate Headquarters, New York
17th Floor Board Room

PART I NEW MACHINE ORGANIZATION CONCEPTS

10:05 INTRODUCTION: Don Dittberner, FSD HQ Plans
GA 46700 ROCKVILLE

a) State objectives of Conference

b) Indicate what is hoped to be accomplished

c) Review Agenda

d) Review hand-out material given to conferees

e) Emphasize need to limit discussion in order to cover full agenda

10:20 ROUND-THE-TABLE INTRODUCTION OF CONFEREES

Each individual identifies himself, his organization, and his background in this area

10:30 COFFEE (Cart)

10:40 THE RW 400 Mr. Jim Batley, FSD Systems Development

a) Brief description of machine specifications

b) Machine organization (flip-chart presentation)

c) Elaboration of the Central Exchange

d) Advantages of the RW 400

e) Weaknesses of the RW 400

12:30 LUNCH

1:30 THE BULL GAMMA 60 — Bill Elmendorf, Research Mohansic

a) Brief outline of machine specifications

b) Machine organization

c) Advantages of the GAMMA 60

d) Disadvantages of the GAMMA 60

e) Comparison to RW 400 concept

2:20 SAGE I and II — Bob Rockefeller, KFSD

a) Features of input and display unit redundancy (backup) concepts

b) Duplexing concepts

2:35 HONEYWELL 800 — Phil Seaman, DS, Competitive Equipment Analysis

a) General machine specifications

b) Multiple sequence counters

c) Comparison to RW 400

2:45 BMEWS - SABER - BMX — Bev Brown, ASDD

3:00 COFFEE (Cart)

3:15 IBM FRANCE STRIDA II — Jim Selfridge, KFSD

a) General Specifications

b) Organization

c) Comparison to RW 400

d) Motivation for design - advantages

3:30 A MICRO-PROGRAMMED COMPUTER — Ken VanMechelen, FSD Owego Engineering

3:50	OTHER RELEVANT SYSTEMS		Don Dittberner, FSD HQ Plans
	a)	RCA 601	Burt Lappell
	b)	Burroughs Modular	Don Dittberner
	c)	LARC	
	d)	Fieldata Accelerator	Jim Batley
4:10	DISCUSSION OF HARDWARE SYSTEMS NOT MENTIONED		Floor
4:30	ADJOURN until 9:30 AM Thursday		

Thursday, May 19th
17th Floor Conference Room

PART II — FEATURES AND ADVANTAGES

9:30 WHY MULTIPROCESSING IS IMPORTANT TO FSD — Tom Gorman, FSD Marketing, Manager, Control Systems

a) Market pressures

b) Outline advantages offered to the military

c) Stress real-time control applications as immediate motivation

10:00 ADVANTAGE AREA CHART INTRODUCED — Don Dittberner

a) Unit Redundancy - or achieving back-up for emergencies

b) Unit Standardization - achieving low unit cost, and simplified logistical and maintenance support

c) Modularity - achieving optimum system configuration and growth

CHARTS * d) Program Compatibility - achieving reduced total investment in programs over a long time period

e) Multi-Programming - achieving maximum utilization of all system components, and processing multiple tasks simultaneously and efficiently

10:10 COFFEE (Cart)

10:20 *UNIT REDUNDANCY — Jim Batley - FSD - SDD

10:50 *MODULARITY — Jim Batley - FSD - SDD

11:10 *UNIT STANDARDIZATION

11:20 *PROGRAM COMPATIBILITY — HOW FEATURES AFFECT SYST. MACH. ORGANIZATION — Bob Bemer, Corporate, Adv. Progress Research

11:50 *SIMPLIFYING PROGRAMMING — John McPherson, Corporate V.P.

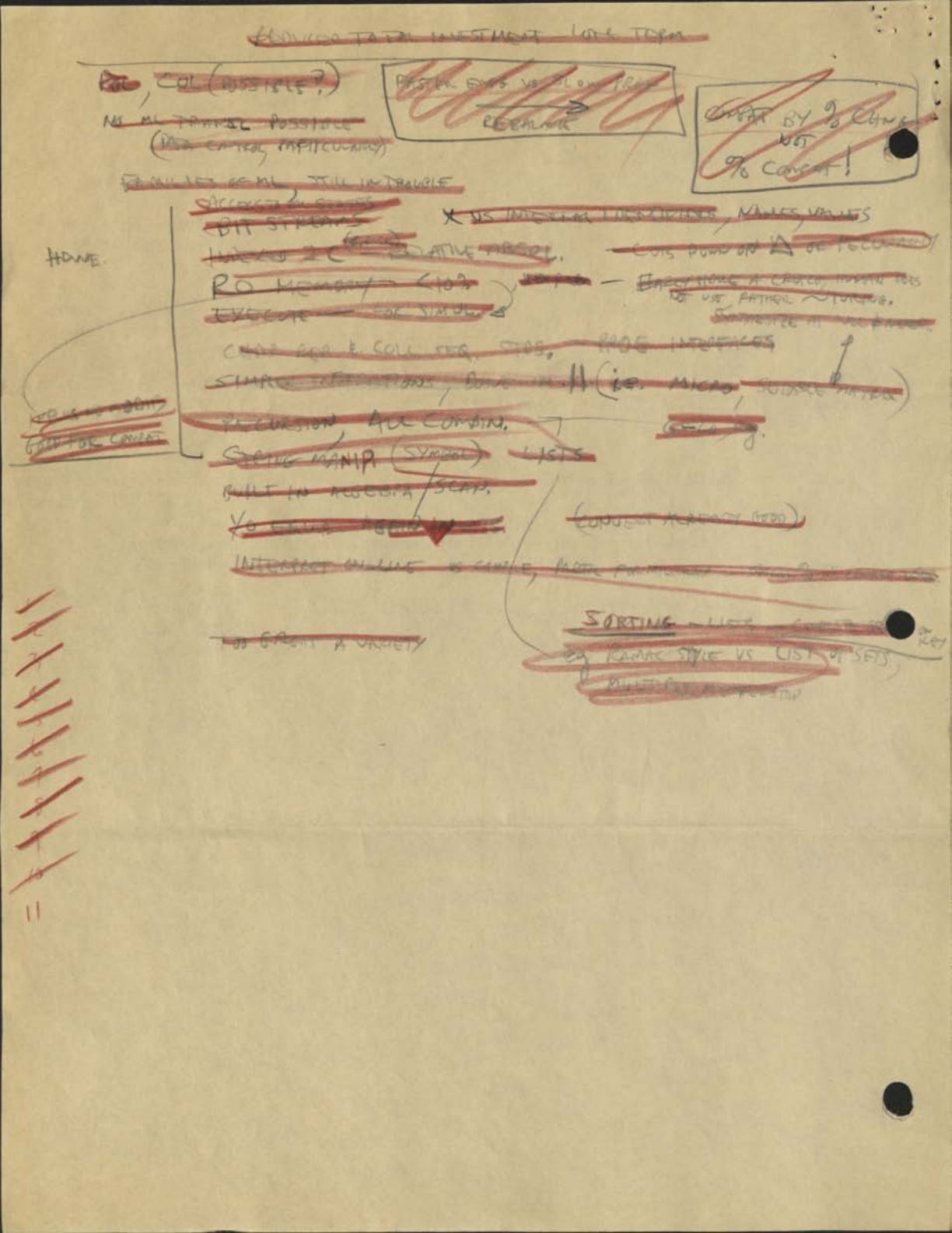
* Format for Discussion of Advantages:

a) Definition
b) Methods of achieving (examples taken from machines presented)
c) Problems in achieving
d) Present IBM equipment deficiencies in achieving
e) Possible feasible method to achieve for IBM

12:20	LUNCH	
1:30	MULTI-PROGRAMMING	Ed Lowry, Data Systems, STRETCH Group
2:00	SOME NEW CONCEPTS	Dr. John Griffith, Corporate Research, WHQ
2:15	IBM PROGRAMMING STATE OF THE ART A COMMENTARY	Miss Elaine Boehm, Corporate Research, Mohansic
2:30	COFFEE (Cart)	
2:35	GROUP DISCUSSION OF ADVANTAGES	J. B. Jackson, Discussion Leader
3:00	EVALUATION OF CONCEPTS	J. B. Jackson, Director of Plans FSD HQ (Discussion)

Documenting group evaluation, generation of documented conference conclusions for distribution to conferees for comment and further study.

4:15	CORPORATE PRODUCT PLANNING SUGGESTIONS	Walt Johnson, Corporate Product Planning
4:30	CLOSE OF CONFERENCE	



MARQUETTE UNIVERSITY

5th Machine Records Conference

APRIL 25, 26, 27, 1960
MILWAUKEE, WISCONSIN

GENERAL SESSIONS Room 206-7-8	STUDENT RECORDS SESSION—Room 206	BUSINESS SESSIONS—Room 208	COMPUTING SESSIONS—Room 204
Monday, Apr. 25 8:15-9:00	**GENERAL SESSION—Welcome to Marquette,** Very Rev. E. J. O'Donnell, S.J., President, Marquette University Rms. 206-208 Program Preview, R. S. Zielinski, Supervisor IBM Tabulating Department, Marquette University.		
9:00-10:00	Verner Ekstrom—Univ. of Wisconsin REGISTRAR, SELF-STUDY	Percy Baker—Mississippi State College PAYROLL PROCEDURES.	James Snyder—University of Illinois THE ORGANIZATION, CONTENT AND USE OF THE ILLIAC AND IBM 650 LIBRARIES AT ILLINOIS.
10:00-10:30	COFFEE BREAK	COFFEE BREAK	COFFEE BREAK
10:30-11:30	Leo M. Corbaci—Univ. of Notre Dame USE OF THE LABELING MACHINE FOR POSTING PERMANENT RECORDS.	Frank Teske—Virginia Pol Institute PERSONNEL RECORDS	Robert L. Smith—Texas A&M College COMPUTING CENTER EXPERIENCE IN UNIVERSITY MACHINE RECORDS WITH THE IBM 704.
11:30-12:00	**GENERAL SESSION**—Preliminary Problems Discussion for Session #4 (2:00-3:00).		
12:00-1:00	**LUNCHEON**—Ballroom, Brooks Memorial Union.		
1:15-2:00	Ronald Neupert—Washington Univ., St. Louis TABULATED REPORTS TO SECONDARY SCHOOLS.	James Byrum—University of Colorado BUDGET AND BUDGET CONTROL.	R. W. Bemer—I.B.M. Corp. World Hdqtrs. PROGRAMMING DEVELOPMENTS IN DATA PROCESSING [illegible] COMPILERS.
2:00-3:00	**GENERAL SESSION**—PROBLEMS WORKSHOP—Bring your problems for general discussion, comments and suggestions.		
3:00-3:30	MORE COFFEE	MORE COFFEE	MORE COFFEE
3:30-4:30	George Crockatt—Wayne State Univ. WE THINK WE LIKE RED TAPE FLEXOWRITER FOR ADMISSIONS AND STUDENT RECORDS.	Francis Finn—Purdue University STORES CONTROL.	Enders Robinson—Univ. of Wisconsin FRONTIERS IN NUMERICAL ANALYSIS.
4:30-5:00	**GENERAL SESSION**—Program Announcements, Bus Schedule, etc.—R. S. Zielinski.		
7:00	**DINNER**—Ballroom, Brooks Memorial Union.		
Tuesday, Apr. 26 8:30-9:30	Romine Matthews—Univ. of Minnesota IBM REGISTRATION PROCEDURES.	Casey Jones—Washington State Univ. THE USE OF PUNCHED CARD SYSTEMS IN HOUSING AND FOOD SERVICE ACCOUNTING.	Harry Rymer—Northwestern University CURRICULUM AND RESEARCH IN COMPUTER SCIENCES.
9:30-10:15	Robert Moors—Univ. of Massachusetts APPLICATION OF THE 1401 IN A UNIVERSITY.	J. B. Combs—University of Missouri ATHLETIC TICKET ACCOUNTING.	Roy Reeves—Ohio State University ADMINISTRATION OF PROGRAMMING EFFORT.
10:15-10:45	COFFEE AGAIN	COFFEE AGAIN	COFFEE AGAIN
10:45-11:45	Miss J. Williams—Wash. State Univ.	Edward Haislet—Univ. of Minnesota	John Hamblen—Univ. of Kentucky

GENERAL CHAIRMAN:
R. S. Zielinski
Supervisor
IBM Tabulating
Marquette University
Milwaukee 3, Wis.

PROGRAM CHAIRMAN—Student Records:
Jacke Feise
Supervisor
Data Processing
Washington State University
Pullman, Washington

PROGRAM CHAIRMAN—BUSINESS:
William S. Dye, III
Manager
Data Processing Service
Pennsylvania State University
University Park, Pennsylvania

PROGRAM CHAIRMAN—COMPUTING:
A. Wayne Wymore
Director
Numerical Analysis Lab
University of Arizona
Tucson, Arizona

1961 CONFERENCE

The 1961 Machine Records Conference will be held at the Pennsylvania State University, University Park, Pennsylvania. Mr. William S. Dye, III, Manager, Data Processing Service, will be the General Chairman.

The Conference Chairmen are looking forward to creating a program of presentations **which you, the delegates, want.** We rely on your suggestions and comments. If you have a particular procedure, problem or presentation, contact the Chairman concerned as soon as possible. If you would like to be on the program, do not hesitate to offer your services.

LET US HEAR FROM YOU SOON!!

FUTURE CONFERENCES

The site of the 1962 Machine Records Conference will be chosen at this year's conference. The conference chairman, and past chairmen, do not designate an institution to host a conference. **We rely on an invitation.** If your institution is willing to host a future conference, please present the invitation during the 1960 conference.

MARQUETTE UNIVERSITY

ANNOUNCES

the

FIFTH ANNUAL MACHINE RECORDS CONFERENCE

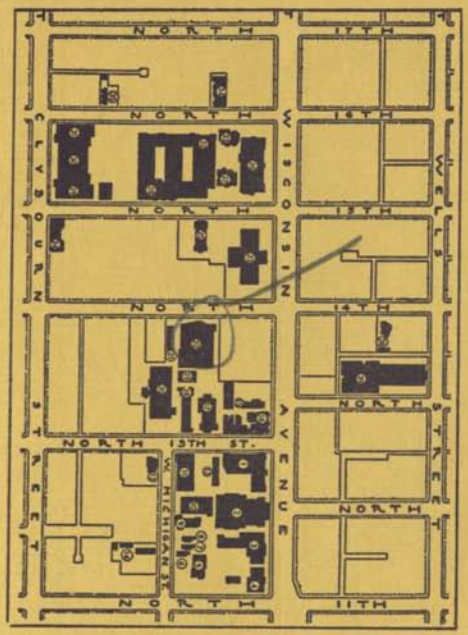
for

EDUCATIONAL INSTITUTIONS

APRIL 25, 26 & 27, 1960

MILWAUKEE, WISCONSIN

MARQUETTE UNIVERSITY CAMPUS



KEY TO BUILDINGS

1. Science Building
2. Gesu Church
3. Johnston Hall
4. Sensenbrenner Hall
5. O'Hara Hall
6. Drexel Lodge
7. Lalumiere Lodge
8. Jogues Hall
9. Regis Hall
10. Bellarmine Hall
11. College of Business Administration
12. 13th Street Annex
13. Copus Hall
14. Hughes Hall
15. Hopkins Hall
16. Grandmora Apartments
17. Carpenter Hall
18. Noonan Hall
19. Science Annex
20. Gesu School Hall
21. Student Affairs Building
22. Brooks Memorial Union
23. Memorial Library
24. Residence Hall
25. Bonifas Hall
26. College of Engineering
27. School of Speech
28. School of Medicine
29. Army ROTC
30. Gymnasium
31. Navy ROTC
32. School of Dentistry
33. Athletic Department
38. Spaulding Hall
39. Walter Schroeder

RWB

WASHINGTON CHAPTER

of the

ASSOCIATION FOR COMPUTING MACHINERY

Dinner Meeting

--- "BEAT THE PANEL TO BITS" ---

The Chapter is fortunate in being able to present an exceptional panel of experts who will evaluate the current status and future position of such automatic programming techniques as ALGOL, COBOL, etc. A discussion and question period will follow the panel's presentation of --

Subject:	"Common Languages for the Uncommon Folks"
Speakers:	Robert Bemer, IBM Joseph Cunningham, USAF Betty Holberton, DTMB Joseph Wegstein, NBS
Date:	Thursday, April 21, 1960
Place:	The Broadmoor, 3601 Connecticut Avenue, N. W.
Time:	Cocktails - 6:00 P.M. Dinner - 7:00 P.M. Speakers - 8:00 P.M.
Tariff:	$3.50 for Dinner
Reservations:	Call Loret Miller - RE-7-6227 (Please make your reservations early !!)

615 PENN AVE
MEET GASS AT SPACE CENTER <5 P.M.

Free Parking at the Broadmoor -- Entrance on Connecticut Avenue

PUBLIC INVITED

PLEASE POST ON YOUR BULLETIN BOARD

NE 157 325 LGA - 440 DCA
CA 272 1010 DCA - 1110 LGA) CAR

REPUBLIC AVIATION CORPORATION

FARMINGDALE, LONG ISLAND, NEW YORK

Telephone CHapel 9-1100

October 22, 1959

40 MEM (30-35)
1-1½ HR
PRINTED MATL?

Mr. Robert Bemer
IBM Corporation
112 East Post Road
White Plains, New York

Dear Bob:

This will confirm our telephone conversation regarding your address to the Long Island Computer Association on November 5, 1959 THURS at 8:00 p.m., the topic being "International Algebraic Language and Commercial Translator Concepts".

I would like to invite you to visit the RAC installation before dinner. If this is convenient, Mr. Julius Honig will escort you from IBM, Garden City to the plant. ~~If not, dinner at 6:00 p.m. Mr. Honig will expect you at IBM, Garden City anytime before 6:00 p.m.~~

I look forward to your visit.

Sincerely,

A. Finerman

A. Finerman
Manager, Digital Computing and Data Processing Division

AF/kc

500 FRANKLIN AVE

NORTH STATE PKWY OR L.I. XPRESSWAY — N
MINEOLA AVE
WILLIS AVE - MINEOLA

X ST FROM DOUBLEDAY, ON CORNER

IV. SELECTED REFERENCES

COMMITTEE

Raymond Walch, Program Chairman
Dorothy Roper
Ella Williams
Norine Kennedy
Robert Downing
Elliot Pierson
Leonard Launer
Ella Rice

OFFICERS

President - Ella E. Rice
Vice President - Helen D. Moore
Secretary - Josephine Tansey
Treasurer - Raymond Walch

Executive Members:
Mary K. Tulock
Harriet Gesler
Henry Syer

V. ACKNOWLEDGEMENTS

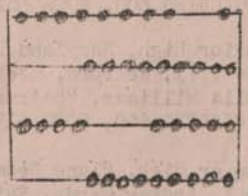
Westport Board of Education
Staples facilities

Staples Service Club
Student Guides

Natalie Lund
Staples Cafeteria Director

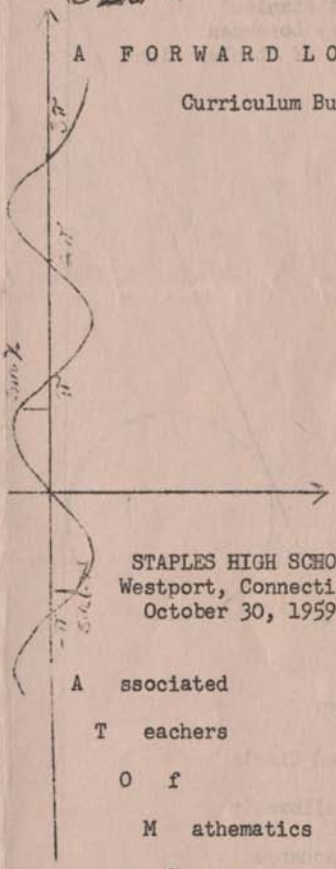
Kathy Huck - Organist

Ann Moniuk - Typist



ATOMIC

CEA CONVENTION



A FORWARD LOOK

Curriculum Bulletin No. 1

STAPLES HIGH SCHOOL
Westport, Connecticut
October 30, 1959

A ssociated

T eachers

O f

M athematics

I n

C onnecticut

TABLE OF CONTENTS

CONNECTICUT EDUCATION ASSOCIATION
21 OAK STREET
HARTFORD 6, CONNECTICUT
Telephone JAckson 2-2159

November 10, 1959

Dear Dr. Berner,

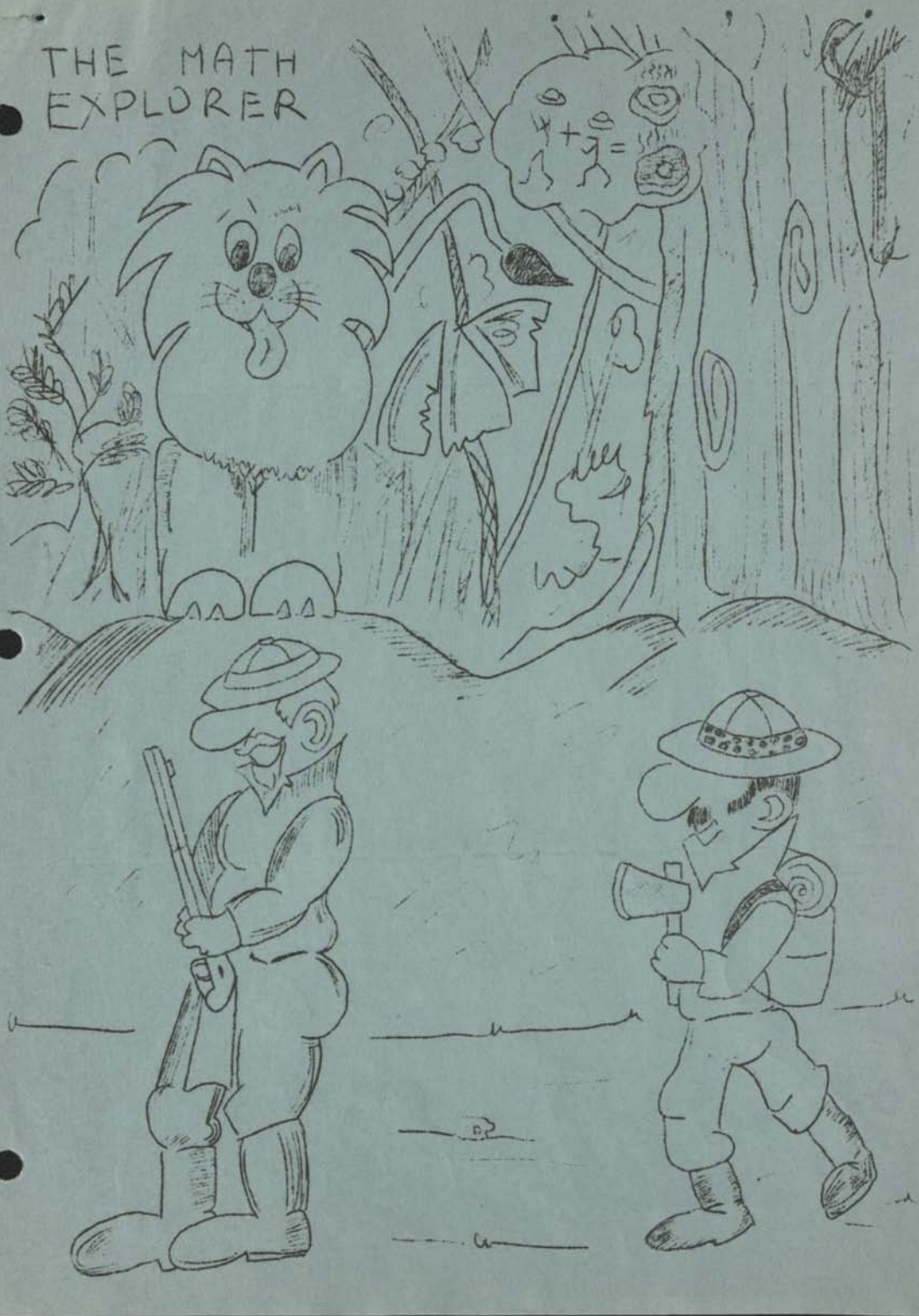
The CEA and the Associated Teachers of Mathematics in Connecticut wish to express appreciation for your participation in our October 30 Convention program. I hope you will meet with the mathematics teachers again for they were most enthusiastic about you and your presentation.

Many thanks for giving of your time in your busy schedule.

Sincerely,

Ella E. Rice,
President - A.T.O.M.I.C.

THE MATH EXPLORER



THE MATH EXPLORER

VOL. 1 No. 3

THE MATH EXPLORER is published monthly by the 7th GRADE MATH SOCIETY of the Weston, Conn. JUNIOR HIGH SCHOOL.

THE OCTAL SYSTEM--COUNTING BY EIGHTS

By Robert H. Bemer
IBM Engineer

It is quite true that most high-speed computers perform arithmetic in the binary system. This is because the symbols required (0 and 1) are conveniently represented by "off" and "on" states of tubes, relays, or magnetic cores. This is very similar to the two states of an electric bulb, as you switch it "off" and "on." However, the user of the computer does not use the binary system for the practical purposes of checking out his problems or telling the problem to the machine (programming). He uses the octal system, which is very closely related to the binary system.

This is a very good reason for this choice and it is linked to the number of symbols a human can easily remember and use. The multiplication and addition tables in the binary system are very easy to remember but the numbers become very large and are hard to manipulate. The decimal system, on the other hand, has a very difficult multiplication table to learn--as you know. Suppose, for example, that we worked in the base 100 instead of 10. That means we would have 99 differently shaped symbols to remember instead of just the 10 symbols 0 through 9. Numbers would be much shorter but it would be almost impossible for even great geniuses to memorize the multiplication table the way everyone does in school today.

Obviously a compromise is needed and the octal system fits this--using eight symbols 0 through 7, almost as many as the decimal system. The important difference is that the octal digits 0 through 7 can be represented as groups of three binary digits. The following table shows this:

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Now let's take a number (this year, for example)--1959, and write it in both the binary and

and octal systems:

Binary	011	110	100	111
Octal	3	6	4	7

Note that we grouped the binary number in three's and used the first table to decode these groups and write the number in the octal system. Although it took 11 bits (binary digits) to express 1959 in binary, it took only 4 octal digits to express the same number, no more than it would in the decimal system. This is the way the computer operator and planner (programmer) reads the numbers from the lighted display on a giant computer. There are even small desk calculators made to operate in the octal system.

Note that you can convert from octal to decimal much easier than from binary. Our number 3647 can be converted by placing below it powers of 8 (rather than powers of 10 as used in the decimal system.)

3	3	4	7
512	64	8	1(powers of 8)

Instead of saying:

1959=1(thousand)+9(hundreds)+
5(tens)+9(ones)

We would say:

3647=3(512's)+6(64's)+4(8's)+7(1's)

If you would multiply this out and add together you would see that it comes out to 1959 in the decimal system.

This usage of the octal system effectively squelches the idea that people used to have, before computers, that the duodecimal system (base 12) would be good. To be effective in our civilization, a number base must have a power of two because this is the cheapest way of transmitting information whether inside of a computer or being sent thousands of miles away by telephone or microwave.

Editor's Note: We would like to thank Mr. Bemer very much for his most interesting article. We are honored to have an employee of IBM take some of his valuable time to write an article for our periodical.

PROBLEMS FOR A RAINY DAY IN THE SPRING
Edited by Vicki Brown

1. Albert is an overgrown ant. He lives in a large ant house. His chamber is circular in shape with an area of .7850 sq. in. Albert wants to throw a big party but he doesn't know how many ants he can fit into his chamber. He does know, however, that he covers a total area of 1/8 of a sq. inch, which is twice as large as an average ant. Since Albert is a poor mathematician, please find out how many average ants he can have to his party. When you arrive at an answer, subtract 1/4 of that from the original answer to allow breathing space.

2. Elmer the Bookworm is a perky little worm who lives in the Burwell's attic. Elmer has a problem and wants you to figure it out. He loves books; in fact they're his favorite dish. For his Sunday dinner he has chosen the Webster's Library Dictionary. But he found out from last Sunday that if he eats 1,889 cu. in. he will get an agonizing stomach ache. If Elmer

eats the dictionary, which is 11 in. high, 12 in. long, and 9 in. wide, will he get a stomach ache?

3. Joe the grasshopper wants to buy a new hat. Joe's head is a trapezoid. One base is 21 inches long, the other base is 19 inches, the sidesboth 25½ inches, and the height 17 inches. The sizes of hats are: small, 97"-100"; medium, 101"-105"; large, 106"-110"; and extra large, 11"-120". (Hint; to find size, find the perimeter of the top of Joe's head) What is Joe's size?

4. Chauncy the Cricket has a problem. He wants to have the outside of his cage painted gold, but Scooge, who owns him, says he can't unless he figures out how much it will cost. But poor Chauncy never learned how to find the area! So let's help him out. The floor is 6 in. by 6 in. The bars are one inch thick and one inch apart.on the walls and ceiling. The cage is 4 in. high. One small can of gold paint costs 50¢. H ow much will it cost to paint the cage, if one can covers 15 sq. in.?

A cute seventh grade girl taking a test had this problem put to her: If a man buys an article for $10.25 and sells it for $8.75, does he gain or lose by the transaction?

The classical answer was given, "He gains on the cents but loses on the dollars."

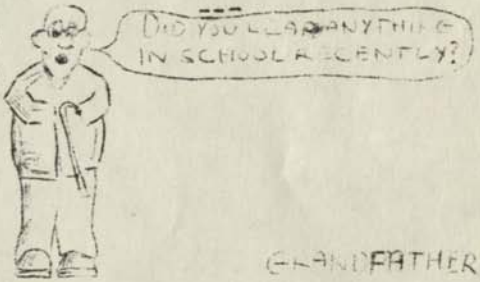
The Line
By Sylvia Robinson

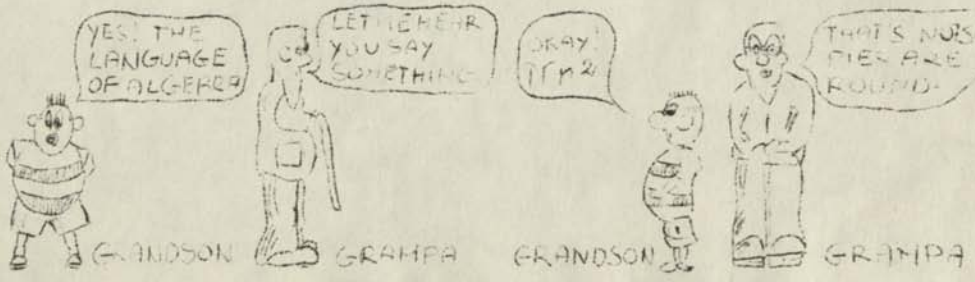
Our intimate friend the line
Exists only in the mind
For it has just one dimension
Which is, of course, extension.
For the line
Is just a sign
With thickness, width, and length
So again I must mention
In order to have <u>just</u> extension,
The wonderful line
Can only be inside the mind.

CONTEST! SUMMER RECREATION!

Here is a problem to think about this summer. "How many different ways can you make change for one dollar?"

An appropriate prize will be presented to the student who thinks of the most ways. At the top of your entry give the total number of ways, and then list each way in chart form. All entries should be given to Mr. Pierson the first week of the new school year.





THE METRIC SYSTEM
By George Colacicco

Nearly two hundred years ago French scientists devised a method of measuring and weighing called the "Metric System". This name comes from the word "meter", the principal standard unit of length in this system. The meter resulted by taking the length of one ten-millionth of the distance between the poles and the equator along a meridian. Many countries in the world adapted this system because of its simplicity except the English-speaking countries which use the English System of Weights and Measures. But even in these countries, the only civilized ones which do not use the Metric System, scientists usually use the metric system because of its accuracy. Since modern transportation brings foreign countries closer to us, it might be wise for an intelligent citizen to learn both systems. Some G.E. engineers discuss the fact. One says: "The tremendous growth of Eastern powers, with their progressive penetration of tools, machines, equipment and raw materials markets, may well put English speaking countries with their antiquated measuring system at a disadvantage. The simplicity that goes with the Metric System saves time, eliminates errors and facilitates comprehension."

All standard units in the Metric System have a scale of relations based on the Decimal System of numbers. The scale of multiples and subdivisors is ten. A uniform system of names is one advantage to the Metric System. All units of weights and measures have the following prefixes: milli - 1 thousandth, centi- 1 hundredth and deci - 1 tenth, are for the subdivisors and deca - ten, hecto - 1 hundred and kilo - 1 thousand are for the multiples. All units get their names by adding these different prefixes to the chief unit. Example: meter and decimeter. The unit of length is the meter, for long distances the kilometer (1000 meters) is used. The unit of surface is the square meter, the multiples and subdivisors of which go by the square of ten, one hundred. One square decameter equals 100 square meters. In large estates measures like ares (100 square meters) and hectares (10,000 square meters) are used. Countries are measured in square kilometers.

The unit of volume is the cubic meter, the multiples and subdivisors of which go by the cube of 10, which is 1000. One cubic decameter equals 1,000 cubic meters. When the cubic meter is used to measure volume of wood, it is called a "stere". The unit of capacity is the liter, it is used for liquids such as milk and wine. All bigger containers have their liquid measured in hectoliters (100 liters). The unit of weight is the gram. One thousand cubic centimeters which have the capacity of one liter weighs one kilogram (1,000 grams) a more widely used standard of weight. For heavy articles the metric ton (1,000 kilograms) is used. In fine scientific problems, the micrometer, a unit of weight, is used. It corresponds to a millionth of a gram. This confirms the accuracy of the Metrick System. To weigh precious stones the carat (k) has now been standardized to 200 milligrams $(\frac{200}{1000}$ of a gram)

Problems:

Example: If John buys 3500 grams of flour, how much will he pay if the flour costs 63½¢ per kilogram

Solution: (1 kilogram = 1000 grams)

$$\begin{array}{r} 3\frac{1}{2} \\ 1000\overline{)3500} \\ 3000 \\ \hline 500 \end{array}$$

$$\begin{array}{r} .635 \\ \times\ 35\text{kg} \\ \hline 3175 \\ 1905 \\ \hline 22225 \end{array} - \$2.22$$

1. Jeff can jump 1 meter and 15 centimeters and little Steven can jump 23 centimeters. How many centimeters less than Jeff does Steve jump?

2. Harold is sent to buy $3\frac{1}{2}$ liters of milk. The old milk store has a small container which holds 2 centiliters. How many times does the milkman have to fill and empty his small container to fill up Harold's

---

FAMOUS MATHEMATICIANS
By Barbara Range

This poem is designed to help you
  remember
Famous mathematicians, if 'ere you
  run asunder.
So please bear in mind as this poem
  winds its way
The different math greats of
  yesterday.

Karl Gauss was the man when only
  ten
Found a short way of adding, and
  then
Pulled on his teacher this little
  joke,
It's a wonder that poor man didn't
  choke.

Urbain Leverrier, born in the
  eighteen elevens,
Developed a theory about the heavens.
He predicted the planet Neptune's
  discovery
Which brought him fame and idolatry.

Charles Peirce, no doubt with the
  help of his dad,
Set up, at twelve, his very own
  lab.

He was the first who used the
  the light wave
As a standard of measurement,
  time it did save.

Sir Issac Newton invented the
  calculus
Which is an important part of
  mathematicus.
A reflecting telescope for
  investigations
Was another one of his important
  inventions.

Euclid, "The Father of Geometry,"
Lived 2,200 years ago in the
  reign of Ptomley.
His "Elements" has lasted through
  the ages
And taught much in its pages.

This poem now ends, its job is
  done.
I hope it has not bored you but
  given you fun.
And now please remember as it
  goes its way,
The different math greats of
  yesterday.

Answers to "Problems For a Rainy Day in the Spring"

1. 9 ants
2. No, Elmer will not receive a stomach ache. The Webster's Dictionary is 1188 cu. in.
3. 101 inches, medium size
4. $3.50

"Metric System" Answers

1. 92 cm.
2. 175 times

Applied Programming
DSDHQ
October 7, 1959

SEMINAR--PROGRAMMING LANGUAGES

For your information I am enclosing a tentative program for the forthcoming seminar in programming languages. Your initial invitation stated that 20 to 25 IBM'ers would attend. However, there has been such an enthusiastic response that we now expect an attendance in the neighborhood of 50. The meeting will be held in the Roger Smith Hotel in White Plains starting promptly at 9:30 a.m. on 13 and 14 October.

The speakers have been asked to limit their talks to about 30 minutes, thus leaving time for a general question and answer period following each talk. If there is time at the end of each day an open discussion on the topic "Programming Systems of the Future" will be held.

tbg/ep

T. B. Glans
Programming Systems

Those to Whom Invitations Have Been Sent:

T. Arbuckle
J. W. Backus
M. P. Barnett
J. C. Batchelder
R. W. Bemer
R. O. Blanchard
W. G. Bouricius
W. Buchholz
S. Campbell
W. Carter
R. L. Cline
G. Collins
L. N. Eselson
C. E. Diss
S. W. Dunwell
P. C. Gilmore
T. B. Glans
R. Goldfinger
B. Gordon
J. Green
W. P. Heising
L. E. Kanter
S. L. Lida
I. C. Liggett
J. R. Lourie
R. E. Merwin
D. Mordy
J. C. McPherson
R. K. Ridgway
N. Rochester
D. W. Pendery
B. L. Sarahan
S. Schmitt
W. M. Selden
R. M. Shapiro
S. J. Szabronski
R. B. Talmadge
J. M. Terlato
D. L. Thompson
F. A. Williams
A. S. Wolf

PROGRAM

SEMINAR--PROGRAMMING LANGUAGES

Tuesday, October 13, 1959

9:30-10:30 R. W. Bemer, DS Programming Systems MODERATOR
Introductory Remarks
Report on standardization in programming, I/O equipment, character sets, etc

10:30-10:45 Coffee

10:45-11:30 W. M. Selden, DS Programming Systems
Report on activities of CODASYL (DOD) Short-Range Task Force

11:30-12:15 R. Goldfinger, DS Programming Systems
Report on activities of and plans of CODASYL Intermediate-Range Task Force

12:15-1:30 Lunch Period

1:30-2:15 B. Gordon, DS Programming Development
Commercial Translator Language and Processors

2:15-3:00 J. Green, DS Programming Systems
ALGOL Language and Processor

3:00-3:15 Coffee

3:15-4:00 I. C. Liggett, ASD Programming Research
Plans for ASD Programming Research

4:00-4:45 W. P. Heising, DS Programming Development
SHARE 709 System (S.O.S)

If time permits, the floor will be thrown open for general discussion of the topic "Programming Systems of the Future."

Wednesday, October 14, 1959

9:30–10:30 J. W. Backus, Corporate Programming Research
Activities in Programming Research

10:30–10:45 Coffee

10:45–11:30 G. Collins, DS Mathematics and Applications
A General Compiler Language

11:30–12:15 N. Rochester, Research
Recent work at MIT

12:15–1:30 Lunch Period

1:30–2:10 M. P. Barnett, MIT
A Language Stylization Transliterator

2:10–2:50 J. R. Lourie, DS Programming Systems
Genesis--Problem Organization Processor

2:50–3:05 Coffee

3:05–3:45 F. A. Williams, DS Programming Systems
The Automation of a Computer Facility through Program Control

3:45–4:25 L. N. Eselson, DS Programming Systems
Machine Independent Assembly System (MIDAS)

4:25–5:00 General Discussion:
The effect of programming systems on machine design and how to achieve effective controls.

* * * * *

Frank Williams

Hardware – Tapes should be able to [illegible] any channel [illegible]
to A CHANNEL

Chair Set - Berner, Ed Lam, Bucholz

BILLS
CHECKS
PAY BANK
AUTO INS.

SEMINAR - PROGRAMMING LANGUAGES

Welcome to the seminar that just grew for lack of a more formal means of communication within IBM and the speed of development in our field.

Sudden resurgence in techniques, particularly due to university research, gets us out of our doldrums. Aim of this conference is tp find otu how to take advantage of our own products. Suddenly we find th computer to be a symbol manipulator - having submerged ourselves morecompletely in the break where computer processing - on the interior - talks its own language, we find better logistic techniques, LSIT processors, external to internal identifiers, etc.

STANDRADIZATION - ALGOL area well known to most, ICIP relationship, Commun-Copenhagen Bulletin. Nov meeting, Jan publish Mark I after considering improvements. Much of our work in IBM aimed toward this. Perlis algorithm swap meeting 15Oct. parallel to European effort. Bauer and Taschenbuch, algorithms and handbook. Sample or typical scan and proceesor decoder. Wrassle at ACM council meeting, decision that Perlis committee would be revised for permanent mainetance of specs and for bus langiage as well.

Business language - CODASYL by DOD. ICT effort. Extreme urgency that appears in users for too late. AiMACO. Specs written in contracts. More complex requirements for business language, little understood by ALGOL boys, etc. Separate data dscription, flow algebra, all types of variables (variablevariable possible). Need for realtime elements (example) Bill and Roy will speak on this, some very new developments.

Prospect of Joint SCEINTific Commerica langua ge - Juliens work show a basic syntax, which can be fleshed with jargon, dialect, or what will you. Bauers report in ARF. Need for joint system ubder single operating controls for moe attractive package, no alternation of tape loading, etc. Appeasr to be quite feasible within 2-3 years. 705 processor bulletin, nicerecognition of the concept. IBM's RECOGNITION OF PROBLEM WITH INTERDIV PROG COMMITTEE.

STANDARDS imply more than language and operating system. English is rigorously a series of xxxx concepts under recursive control..Elements may be written or pictorialized symbols, or verbal sounds strubg together. 2 men may know English and yet be unable to communicate. Deafness, different audio range, different verbal sound mapped to symbols. Interesting to construct a differnet mapping that would work just as well, learn to talk fluently in it. Swap sounds of D and F, L for C, differenet diphthongs. Would it still be En lish. Chart for Learson in Basile committee xhows how std languages force standard representation. McP and B committees oth working on this now, likely to have str ng effect. Awakened to consumer pressure. Fieldata example. Paper tape, escape character. Can always compensate for variance but it is costly to processor. Example of (in FORTRAN, versus 8 instructions for 256 unique characters. Delimiting, subscripting, functions and combinationsthereof. Wide uncontrolled variance at present time, Unless we take lead, we show up badly. Justgxcqcx now recouping our prestige in commercial lanagues.

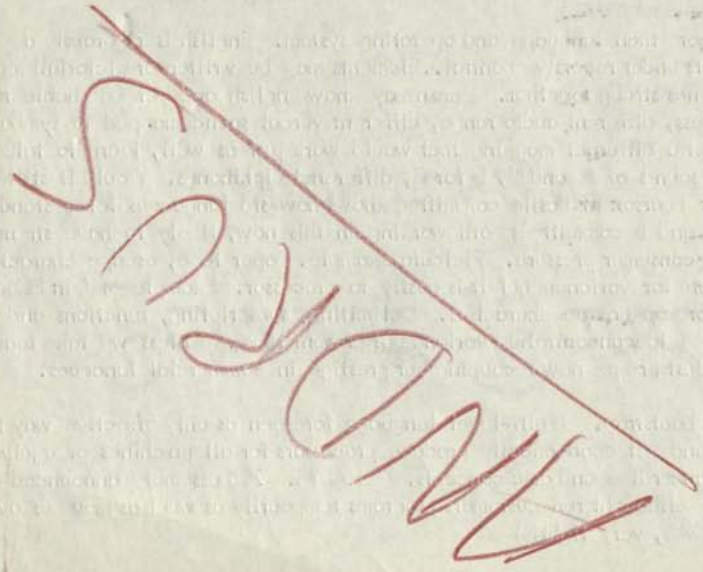
OK world this year is bootstrap. Multi-level language foreseen as only effective way to free machine design and yet economically produce processors for all machines of equivalent intelligence. Muststandardize on basic concepts. CHARTS. 705 assembly announced on 8th 9th. Eeven this, althought run customers programs tempoarily as weel as mobe us over, is npt sufficient. MIDAS, (very lightly)

~~Know we have more than enough people for a legit seminar, but still try to runit like one. Being taped for the record. PS will edit and distribute, for benefit of many who cannot be here.~~ Believe many things to be discussed here are most basic. Tom Glans will caution on tape recording breaks. Also thank him for setting up meeting, but dont know what he means by throwing the floor openfor discussion after. Only one I rememeber was Beverly Hills HS gym that had a swimmingpool underneath. Hope he hasnt pkanne this.

Will have to leave tommorow for NYC, CODASYL with Roy. Would there fore ask (ask before to verofy ----------) to moderate tommowro, particularly as the smoke is likely to get alittle thicker. Often said that people have an easier time with small talent at a smaller company. Simmons EIA, IBMs concentrartion of talent when you try to get a non-iBMer to balance a committee. Graces remark onMCP looking better each t iem a new prospect was metnioned. Lets try to discover ways to produce a product worthy of this powerful talent, not get dissipated thru lack of communcation and planning. HOPE LATER.

THIS INFORMATION CAN BE USEFUL FOR MACHINE DESIGN AS WELL.

Coffee now to get awake for the important elements of the session. Bill Selden speaks right after on CODASYL PDQ (the vulgar title for SHOTRT Range Cpmmittee). Maybe all of you didnt know it but you are effectively here this morning by courtesy of the SELDEn family, since Bills grandfather had the first patent on an automobile. Bill ,I feel, is e qually competent in programing systems.



Applied Programming
DSDHQ
September 24, 1959

SEMINAR--PROGRAMMING LANGUAGES

There has been a great deal of activity recently in trying to standardize on a universal data processing language. The Department of Defense has initiated the studies with the hope that they can utilize a common language, such that programs may be used on any of their present or future machines. A large group of manufacturers in Great Britian have also organized a committee to standardize on a common language.

IBM is currently very active in committee work on the DOD effort. Bill Selden is a member of the Short Range (PDQ) Task Force; Roy Goldfinger is chairman of the language structure group of the Intermediate Range Task Force; and I am participating as an advisor to the Executive Committee on the overall effort.

IBM is also committed to further develop the ALGOL and Commercial Translator languages as well as write processors to translate these languages. Other groups within the company are also engaged in programming language research and development.

In light of this activity I think it appropriate to hold a two-day seminar in the hope of accomplishing the following:

1. Acquainting the various interested parties of the activities of the others
2. Disseminating new techniques developed by individual groups, if of general interest
3. Apprising the group of programming language development outside IBM in this Country and abroad
4. Unifying the position of IBM such that we neither duplicate effort nor show different faces to organizations outside IBM

Programming Systems therefore plans to conduct a seminar in White Plains on October 13 and 14. About 20 to 25 IBM people will be invited, and a number of these will be asked to talk informally about their activities in this field and their plans for the future. Ample time will be allowed for informal discussion. The program is now being worked up and as soon as it is firm I will send you a copy.

Will you please consider this letter an an invitation to attend.

RWBemer

R. W. Bemer, Manager
Programming Systems

tbg/ep

Participants:

T. Arbuckle
J. W. Backus
J. C. Batchelder
R. W. Bemer
W. G. Bouricius
S. Campbell
G. Collins
C. E. Diss
R. Goldfinger
B. Gordon
J. Green
W. P. Heising
I. C. Liggett
J. R. Lourie
J. C. McPherson
N. Rochester
W. M. Selden
R. M. Shapiro
S. J. Szabronski
R. B. Talmadge
A. S. Wolf

August 12, 1959

Mr. Elliot Pierson
109 Cummings Avenue
Fairfield, Connecticut

Dear Mr. Pierson:

I am happy to accept your offer to participate in a panel discussion.

I do this because I am exceptionally interested in education, probably because my father was Superintendent of Schools in Muskegon, Michigan. As such he was very interested in curriculum improvement to keep step with developments outside of the school.

I presume it would be helpful to prepare an abstract of a short individual presentation before discussion. Perhaps we should have a short talk for you to discover my areas of interest and knowledge. My telephone number is Capital 7-6213, or perhaps you would stop in at my home, which is on Weston Road, in Weston just south of the shopping center.

Very truly yours,

rwb/ep

R. W. Bemer, Manager
Programming Systems

FRI OCT 30 — 10 AM — STAPLES SCHOOL AUDITORIUM
DR MAYER, DIR AAAS 1 HR TALK
4 INTROS TO DEMOS IN AFTERNOON
(ME) 30 MIN, START ~ 11 AM., THEN LUNCH?
"EFFECTS OF 'NEW WORLD OF MATH' ON WORLD OF WORK"

August 2, 1959
109 Cummings Ave.
Fairfield, Conn.

Mr. Robert Bemer
Data Processing Division
IBM
112 East Post Road
White Plains, New York

Dear Mr. Bemer:

I wish to thank you for your interest in the math periodical of Weston Junior High. Your article "The Octal System -- Counting by Eights" was the feature of our last edition. Enclosed is a copy of that edition of the "Math Explorer".

It is wonderful to have expert resource people like you take an interest in the teaching of mathematics. At the present time I am serving on a committee which is planning an important meeting of teachers of mathematics in southern Connecticut at Staples High School in Westport on October 30th. I would like to take advantage of your interest in the teaching of mathematics, and ask you to serve on a panel to discuss the "Modern World of Mathematics" at this meeting.

If I receive a note of your acceptance, I will be glad to send the particulars as more detailed plans are formulated.

Sincerely yours,
Elliot Pierson

THE Math Explorer

VOL. III No. 1

THE MATH EXPLORER is published by the MATH CLUB of the Weston, Connecticut JUNIOR HIGH SCHOOL.

STAFF

Editor-in-Chief: Kip Schur
Assistants: Debbie Brenner
Peter Sprague
Ellen Conway
Faculty Advisor: Elliot Pierson

EDITORIAL

During this generation the world has experienced a scientific and technological revolution. The unlocking of atomic energy and the probing of outer space overshadow many of the less dramatic advances. The rapid progress of science and technology in this age can be attributed directly to the development of the electronic digital computer. The large-scale, high-speed, automatic digital computer has made it possible for mathematics theory to be teamed with the computing machine to produce answers that are needed by physicists, engineers and other scientists to accelerate their creative work.

The electronic computer is superior to human calculations for two basic reasons, its speed and greater memory. Computers exist which perform a calculation in nano-seconds (billionths of a second) and can store millions of bits of information. This combination has given scientists a tool that makes quickly and efficiently computations which were formerly completely impossible.

Through their tremendous memory, electronic computers are utilized by businessmen and industrial management as the central part of an entire system of machines known as electronic data processing systems. The essence of business is data: information conveyed by inventory, statistical and financial reports, bills, estimates, etc. The development of high-speed, high-storage capacity, electronic computers gave rise to advancing processing techniques to simplify an avalanche of calculations and paper work necessary for business and industry.

An inkling to the scope of the electronic computer can be seen in the following list of some of the things high-speed computers with large memories are now doing:

1. Simplifying the complicated statistical work of the Metropolitan Life Insurance Company.
2. Developing the scientific data for a nuclear reaction.
3. Keeping track of everything Macy's sells, and printing a list of what needs to be re-ordered every day.
4. Designing the latest missile for Boeing Aircraft.
5. Remembering the payroll for many large companies such as General Electric and printing a check for the proper amount for each employee every pay day.
6. Aiding the U. S. Weather Bureau in weather forecasting.
7. Predicting the sales of General Motors for production control purposes.
8. Keeping track of every ship, tank, plane, can of beans, rifle, uniform, button, etc. that the Armed Services own all over the world.

9. Computing microphotometer for cell analyses in biological research.
10. Keeping tabs on every reservation, passenger, meal and flight of American Airlines.
11. Remembering for a bank how much money each depositor has in his balance, adding whenever he makes a deposit and subtracting every time he cashes a check, and printing a statement every month.
12. Solving pipeline problems for Standard Oil.
13. Checking and evaluating income tax returns for Uncle Sam.
14. Automatic measurement of star positions for astronomers.
15. Formulating minimum cost feed mix for a large dairy farm.
16. Helping to guarantee safety in U.S. space flights.

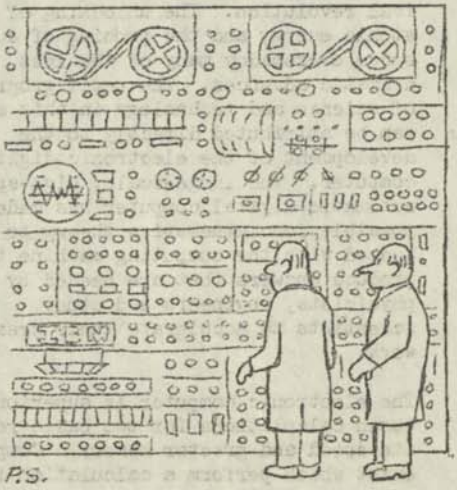
Credit for a very timely and appropriate cover for our "computer" issue is due Kip Schur and Peter Sprague who realize that every space flight is supported by a system of computers. The most critical period in any space flight is the first few minutes. In the first three seconds of flight if every system isn't functioning A.O.K., the astronaut must be rocketed to safety. No group of human computers could possibly receive the data, make the necessary calculations, and transmit the results back to the space vehicle in so few seconds. The electronic digital computer handles the problem with ease.

In only a few years electronic computers have been invented and improved at a tremendous rate. But its basic concept was provided by nature to our earliest ancestors. This simple aid to computation, a digital computer in the strictest sense of the word, can be seen in any kindergarten class where the children count on their fingers. The digital computer, therefore, has evolved from the simple beginnings of numbers, through the abacus and the simple calculators of Pascal, Leibnitz and Babbage. It is the culmination of mathematical thought supported by the entire mathematical family, having been founded on the fundamental concepts of arithmetic, algebra, geometry and trigonometry, and developed through calculus and modern mathematics such as symbolic logic and Boolean algebra.

The electronic digital computer was invented and is being improved as a consequence of man's ingenuity, his imagination and most important, his mathematics. With such a tremendous tool applying mathematics which is affecting and altering our daily lives even if we have little direct contact with it, it seemed appropriate for the "Math Explorer" to devote an issue to its consideration.

We at Weston Junion High were extremely fortunate to have the opportunity to work with an IBM1620 computer under the supervision of Mr. John D'Alessandro, a systems engineer with IBM. Actual contact with this computer aroused a considerable amount of enthusiasm, and many students submitted workable programs to test on the computer itself. Some of these programs may be seen in this issue. A result of our experience with the computer was the formation by the Weston Math Club of a special group exclusively devoted to programming.

We hope you will enjoy this issue.



P.S.

If we only had one to solve the problem $x \cdot y =$ PEACEFUL EXISTENCE

— JEFF POMERANTZ
GRADE 7

THE BINARY SYSTEM AND THE COMPUTER

By Geoffrey Frost, 7th Grade

The Binary system is a system of notation based on the number two. The word binary is derived from the Latin word binarius, meaning two by two. It was devised by Gottefried Wilhelm Von Liebnitz, but was apparently used in China over four thousand years ago.

Since the binary system is based on two instead of ten, values will be, from right to left,

$$1, 2, 2^2, 2^3, 2^4, \text{ and}$$

so on. The corresponding place values in the decimal system are, of course,

$$1, 10, 10^2, 10^3, 10^4, \text{ etc.}$$

Only two figures are used in this system, 0 and 1. Here is a chart comparing base two values and base ten values:

10^1	1	2^3	2^2	2^1	1
	1				1
	2			1	0
	3			1	1
	4		1	0	0
	5		1	0	1
	6		1	1	0
	7		1	1	1
	8	1	0	0	0
	9	1	0	0	1
1	0	1	0	1	0

We now see that the binary system has much longer numbers for numerical quantities than does base ten. This is illustrated very clearly in the translation of 1,000,000 into base two: 11,110,100,001,001,000,000. From this number we may conclude that numbers in binary notation are so large and cumbersome that the binary system would have no practical use; however, nothing could be farther from the truth. With the advent of the computer, the binary system became indispensable. We may now ask ourselves, "What possible connection could the binary system have with the computer?"

Before we answer this question, we must first visualize the computer. Broken down into its basic components, we would see that it consisted of millions of electrical circuits. So let us examine an electrical circuit using a schematic drawing.

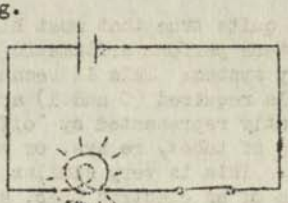


FIG. 1

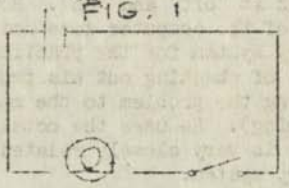


FIG. 2

There are only two conditions of the electrical circuit: Complete, with switch closed (Fig.1) and Incomplete, with switch open (Fig.2). In Fig. 1, the lamp will light; in Fig. 2 it will not.

Because there are two conditions of an electrical circuit, and two figures in the Binary System, one could be substituted for the other. Let us suppose that there is, on a certain computer, a bank of five lights. (Fig. 3):

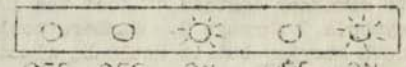


FIG. 3

If we say that a light on stands for a one (in binaries), and a light off means a zero, we could easily interpret the number shown; 101 base two. Translating this to a decimal number, we see that 101 base two equals 5 base ten. Combinations of lights are used to signify different binary numbers, which are then translated into base ten. Most computers translate the binary output into a decimal number, which can be read directly by the operator of the machine.

The binary system is far from obsolete. It is one of the most important aspects of modern electronic computers. Without it, no computer could function, for it would have no language to use in its computations.

THE OCTAL SYSTEM -- COUNTING BY EIGHTS

By Robert H. Bemer
IBM Engineer

It is quite true that most high-speed computers perform arithmetic in the binary system. This is because the symbols required (0 and 1) are conveniently represented by "off" and "on" states of tubes, relays, or magnetic cores. This is very similar to the two states of an electric bulb, as you switch it "off" and "on". However, the user of the computer does not use the binary system for the practical purposes of checking out his problems or telling the problem to the machine (programming). He uses the octal system, which is very closely related to the binary system.

This is a very good reason for this choice and it is linked to the number of symbols a human can easily remember and use. The multiplication and addition tables in the binary system are very easy to remember but the numbers become very large and are hard to manipulate. The decimal system, on the other hand, has a very difficult multiplication table to learn--as you know. Suppose, for example, that we worked in the base 100 instead of 10. That means we would have 99 differently shaped symbols to remember instead of just the 10 symbols 0 through 9. Numbers would be much shorter but it would be almost impossible for even great geniuses to memorize the multiplication table the way everyone does in school today.

Obviously a compromise is needed and the octal system fits this--using eight symbols 0 through 7, almost as many as the decimal system. The important difference is that the octal digits 0 through 7 can be represented as groups of three binary digits. The following table shows this:

Binary	Octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Now, let's take a number (this year*, for example) -- 1959, and write it in both the binary and octal systems:

Binary	011	110	100	111
Octal	3	6	4	7

Note that we grouped the binary number in three's and used the first table to decode these groups and write the number in the octal system. Although it took 11 bits (binary digits) to express 1959 in binary, it took only 4 octal digits to express the same number, no more than it would in the decimal system. This is the way the computer operator and planner (programmer) reads the numbers from the lighted display on a giant computer. There are even small desk calculators made to operate in the octal system.

Note that you can convert from octal to decimal much easier than from binary. Our number 3647 can be converted by placing below it powers of 8 (rather than powers of 10 as used in the decimal system.)

3	6	4	7
512	64	8	1 (powers of 8)

Instead of saying:

1959 = 1 (thousand) + 9 (hundreds) + 5 (tens) + 9 (ones)

We would say:

3647= 3(512's)+6(64's)+4(8's)+ 7(1's)

If you would multiply this out and add together you would see that it comes out to 1959 in the decimal system.

This usage of the octal system effectively squelches the idea that people used to have, before computers, that the duodecimal system (base 12) would be good.

To be effective in our civilization, a number base must have a power of two because this is the cheapest way of transmitting information whether inside of a computer or being sent thousands of miles away by telephone or microwave.

*Editor's note: Mr. Bemer's article is a reprint from the Vol. 1, No. 3 issue of the Math Explorer published in 1959.

CALCULATING GASA

by Peter Maeck
7th Grade

It has been an eventful year. Many revolutionary events have taken place. For one, the son of Commander Alan Shepard, who, as you can easily remember, gave the United States such a boost into space back in '61, has founded the first settlement on the moon, but the moon will most likely not be used for extensive living for awhile. At present there are astronomical specialists there surveying the surface. Another thing, as most have been observing, is the trend in children (with wealthy parents) to play with the new action-filled jet-propelled Halley's Comet Whizzer. This elaborate toy was recently invented and was styled after Halley's Comet which returned two years ago. This fantastic and educational toy whizzes through the air always two feet above the ground. Using insts (components developed to repel objects in a horizontal path), the child is safe from ramming into walls, trees, etc. The only drawback is that it sells for an overwhelming price. In addition to these there have been many other things that have enhanced and forwarded this year of 1988 into being just about the most revolutionary in history.

Now, this November, we are witnessing a climax to this exciting year. This, as most people know, is the candidacy of the first woman contender for the presidency, and, even more stunning, the candidacy of a sensational and highly developed electronic computer as an opponent. The computer is endorsed by International Business Machine and men from both the Republican and Democratic parties. The woman, Mrs. Molly MacGovern, 46, is a Republican. The computer has been nicknamed "Gasa" which stands for "Grasp All, Solve All." This is because of its rapid and complete grasp of any situation and its quick and usually excellent solution. Tonight, the night of November 8, 1988, election night, the votes are beginning to be tabulated by electronic computers. The voting is about tied so far, but before tomorrow is very old we will know the results. Of course, if the computer is elected, a highly trained man will accompany it a as vice-president. This man would be the computer's top advisor and chief programming supervisor. But the American people would still, primarily, be voting for the computer.

This electronic brain is a big favorite in political circles because of its witty and ready comments and answers to ever-inquiring reporters. It has been a great success on television. In fact, the National Broadcasting Company has attempted to obtain possession of it to help them in their as yet unsuccessful attempt to revive TV westerns.

In a television interview the computer made it clear that it was far superior to any human in mental abilities. Although this may be true this type of attitude may get the computer and its supporters into difficulties. This was shown when the computer challenged Dr. Goshen Von Groilenfagel, acclaimed by practically the entire world as the most brilliant person ever to live, to a game of "Olizsestoles". Olizsestoles is a highly concentrative game played only by the most educated and brilliant persons, chiefly because they are the only ones who are capable of understanding it. In any case, the computer defeated Dr. Von Groilenfagel handily and then advised the brilliant doctor to polish up on his "prolistig", which is said to be one of the major sections of the game. With this, Dr. Von Groilenfagel sullenly left the room. He had been playing the game for twenty-three and one-half years.

But now in Election Central in Chicago, it is 10:57 p.m. and the voting is showing varying trends. About ten minutes ago a program was put into one of the computers we have here predicting who will win. Another one was a question asking if the computer at Election Central would like to change places with the computer running for the presidency. In reply, the answering computer typed out, "I have a steady job here."

But now for a few words about the woman candidate. She isn't as witty or imaginative as her opponent, but she does have certain qualities that the computer doesn't have. Molly MacGovern is a human and that helps because as President she would deal with humans. When she met "Gasa" in a heated tele-

vision debate, her opponent typed "Sorry you will lose in November." She had a hard time in making a reply. But in making decisions and in thinking of the country, she does very well - but so does the electronic computer.....

The hours pass and the balloting and tabulations slowly come to an end.

"Yes, here it is! 2:33 a.m. November 9, 1988," says the TV announcer. "The last tallies have come in. We have a new president of the United States, as you can well see on your television screen! The campaigning is over now; the voting has ended. The duel between Mrs. Molly MacGovern and Gasa is through."

Well, who do you think won?

PROBLEMS FOR

"HUMAN" COMPUTERS

Edited by
Debbie Brenner
Grade 9

1. Maggie was carrying a box with "twist" records to her girl friend's house. A young man in a sports car side-swiped her. Maggie was not hurt, but all her records were broken.

"I want to pay you for the broken records," said the young man. "How many records did you have in your box?"

Maggie replied, "I don't know how many records I had. But I remember when I counted them by twoe, there was one record left over. When I counted them by threes, there was one record left over. And when I counted them by fours, there was one record left over. But when I counted them by fives, there were no records left over."

(How many twist records did Maggie have?)

2. A 12-inch Chubby Checker long-playing record has grooves that start a quarter-inch from the outer edge and finish with a last groove having a 5-inch diameter. Assuming there are 120 grooves to the inch, how far does the needle travel in playing the entire record?

3. As John Glenn or any astronaut is aware, the year isn't constant among the planets of this solar system. The number of days in a year on Mercury, Venus and Earth altogether total 10 less than a Martian year. It takes 14 days more than 3 Venusian years to make a year on Mars. The number of days in a year on all four planets is 1362. How many days has a year on Mercury, Venus, and Mars?

4. Herman objected to paying $3 to a blacksmith for shoeing his horse, Elmer. The blacksmith then suggested that he would charge 1 cent for the first nail, 2 cents for the second nail, 4 cents for the next, and double the amount for each succeeding nail he used. Herman agreed to this since poor Elmer needed a shoeing very badly. Only 32 nails were required after all.

Herman was astonished at his bill. What was the amount of the bill?

5. For the human calculator; using the indicated numerals and any mathematical symbols express the indicated numbers:

a) 2 using two 3's.

b) 10 using three 9's

c) 1.5 using three 3's

d) 19 using four 9's

e) 727 using four 6's

(Credit is due Melinda Murray, Grade 8, and Geoffrey Frost, Grade 7, for their contributions to this section.)

GOTRAN - A COMPUTER LANGUAGE

By Kip Schur
Grade 9

Despite many present-day connotations, a computer is not a brain. It cannot think. Without human guidance a computer would not be able to do anything. A computer is only a very complicated set of electrical circuits, which are able to function only when instructed by precise directions.

Gotran is the term applied to a symbolic code used in directing a computer through specified operations. It is one of many different computer languages, and is designed for use specifically with the IBM 1620 Computer.

A gotran program is simply a series of individual commands set up for the computer to follow. These instructions are fed into the computer, and from there the computer does the rest of the work. Gotran statements closely resemble normal mathematical statements, and the easiest way to show this is to analyze an example program. Suppose we have this problem: Write a program that gives the "coupled exponential" sequence -

1^1, 2^2, 3^3, 4^4, 5^5, 6^6, etc.

Have the program print the number and its coupled exponential value for number 1 to 25.

The program for this problem would be:

```
  A   =   1
1 B   =   A * * A
  Print, A, B
  A   =   A + 1
  E   =   A - 26
  IF (E) 1, 2, 2
  2 STOP
  END
```

The following is a step-by-step analysis of this program.

Statement 1. A = 1 This statement simply assigns the value 1 to A. However, the statement does not exactly mean A equals 1. In gotran the "=" sign means "is replaced by", so this statement really means A is replaced by the value 1. (This makes statement 4 possible.)

Statement 2. 1 B = A * * A The double asterisk (**) means "raise to the power of". Thus it gives this statement the meaning B is replaced by the value of A raised to the power of A. (A single asterisk denotes multiplication in gotran programs.) B, of course, is the value obtained by using A as a factor A number of times. Since at this point in the program A has the value of 1, the specific meaning of this statement is B is replaced by 1 to the first power, or $B = 1^1$.

The number 1 at the beginning of the statement is a reference number to the machine. It serves as a means of identifying this statement later in the operation of the program.

Statement 3. PRINT, A, B This statement directs the machine to print the values of A and B as they stand at this point. The machine, at this time, would print 1, 1 because A = 1 and $B = 1^1 = 1$. Later in the program, since the values of A and B will change, so will the printed values of A and B giving the desired table of values.

Statement 4. A = A + 1 This statement assigns a new value to A. The new value is 1 greater than the old; that is, 2. Note that this is an application of "=" in gotran which cannot be used in ordinary math. When the machine comes to this statement again, it would replace 2 as the value of A with the value 3. The cycle would continue as long as the machine keeps returning to this statement.

Statement 5. E = A - 26 The purpose of this statement is to assign a value to E for use in the following statement. This program is to be designed to print the coupled exponentials up to 25^{25}. New values must be assigned to A, through statement 4, until it reaches the value 25. At that point B will equal 25^{25}, which is the last desired exponential in the sequence. In this statement, it is important to determine whether E is positive, negative or zero. If A is less than 26, then it will be negative (if $A < 26$, then $A - 26 < 0$). If A is greater than 25, then A - 26 will be zero or greater than zero. The importance of determining this is to be seen in the explanation of the next statement.

Statement 6. IF (E) 1,2,2 This statement, known as the IF statement, uses three numbers in the statement to denote step numbers, such as the number 1 in statement 2 and the number 2 in statement 7. In every IF statement, three numbers follow the variable in parentheses. The first of the three is the number of the step to which the computer is to go next if the value of the variable (E in this case) is negative. The second number is the reference number of the step the computer is directed to if E is zero, and the third is the step where the computer is to go if E is positive. Basically, the IF statement can be shown structurally as follows:

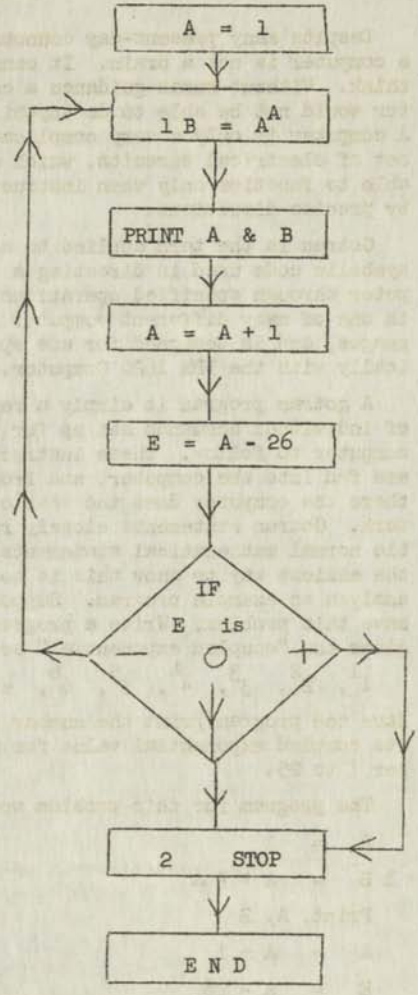
IF (E) (is negative, go to step 1
(is zero, go to step 2
(is positive, go to step 2

The IF statement as used in this program will operate in such a way that the values for A and B (A^A) will not be considered for values of A greater than 25.

Statement 7. 2 STOP This statement tells the computer first to stop any further operations, and second, that the program is finished.

Statement 8. END This statement erases all values of variables used in the program and all the directed operations. In so doing leaves the machine free to assign new sets of values and operations for another program.

Here is an explanation of this program presented in a flow diagram. The direction in which the computer moves from step to step is indicated by arrows. Rectangles denote simple statements or commands; diamonds indicate a decision (IF) statement:



The program for "coupled exponentials" is only one of countless possible gotran programs. Here is another. See if you can follow the reasoning behind it.

Problem: Write a program titled "The Race" to solve the following problem. Consider two sequences, A and B. Sequence A starts at 100 and has a constant growth of 1000 every number. Sequence B starts at 2 and triples with each number. How many steps does it require for sequence B to catch up with sequence A? Program:

```
      A = 100
      B = 2
      C = 0
    1 A = A + 1000
      B = B * 3
      C = C + 1
      D = B - A
     IF (D) 1, 2, 2
    2 PRINT, C
      STOP
      END
```

This particular program has been chosen because it can be followed and computed easily by the reader. It may help to set up a flow diagram. This way it can be seen how the computer moves from one statement to the next, and the reader can follow this same procedure, doing the necessary computations. When this program was tested on a computer, the answer was printed within a very few seconds. It is really not worth the trouble to write such a simple program, since writing the program takes almost as long or perhaps even longer than simply solving the problem by arithmetic. Writing programs for and operating a computer begins to pay off when problems are involved that would take mathematicians months or years to solve without a computer. Suppose "the race" problem were such that it would take, for instance, 750,000 steps for sequence B to catch up with sequence A. It would take a great deal of time for a man to figure this out by himself, but with the aid of a computer the whole process would take a few minutes. This is where the computer really proves its worth.

In future years, man will become more and more dependent upon computers. The day is not far off when man's very survival will rest for a good part on computers.

The study of computers and programming is a fascinating one. Every day it becomes an increasingly important one.

==========

Answers - "Problems for "Human Computers"

1. 25

2. The needle travels only $3\frac{1}{4}$ inches, the distance from the outermost groove to the innermost. It's the record that goes around in circles, not the needle.

3. Earth 365 days; Mars 686 days; Venus 224 days; Mercury 87 days.

4. \$42,949,672.95. The sum of the series $1+2+4 = 7$; $1+2+4+8 = 15$; $1+2+4+8...+2^n = 2^{n+1} - 1$.

   $\therefore 1+2+4+8...2^{31} = 2^{32} - 1 =$

   $4,294,967,296 - 1.$

5. a. $\frac{3!}{3}$

   b. $9 + \frac{9}{9}$

   c. $\frac{3 \times 3}{3!}$

   d. $9 + 9 + \frac{9}{9}$

   e. $\frac{6}{6} + 6! + 6$

THE BRITISH COMPUTER SOCIETY

Conference Dinner

KING'S COLLEGE
CAMBRIDGE

Wednesday, 24 June 1959

CONSOMMÉ JULIENNE

Rudesheimer
Berg Burweg
1953

PETITS PÂTÉS DE CREVETTES

Pommard 1953

CANETON RÔTI

POMMES NOUVELLES

PETITS POIS

SALADE D'ORANGE

CRÊME BRULÉE

KISH AU PARMESAN

DESSERT

Sandeman 1945
Old East India Madeira

CAFÉ

TOASTS

THE QUEEN

THE GUESTS

Proposed by
D. W. Hooper, Esq

Response by
Dr R. W. Hamming
President, Association for Computing Machinery

Georgetown Road
Weston, Connecticut
April 16, 1959

Dear Mr. Beiler,

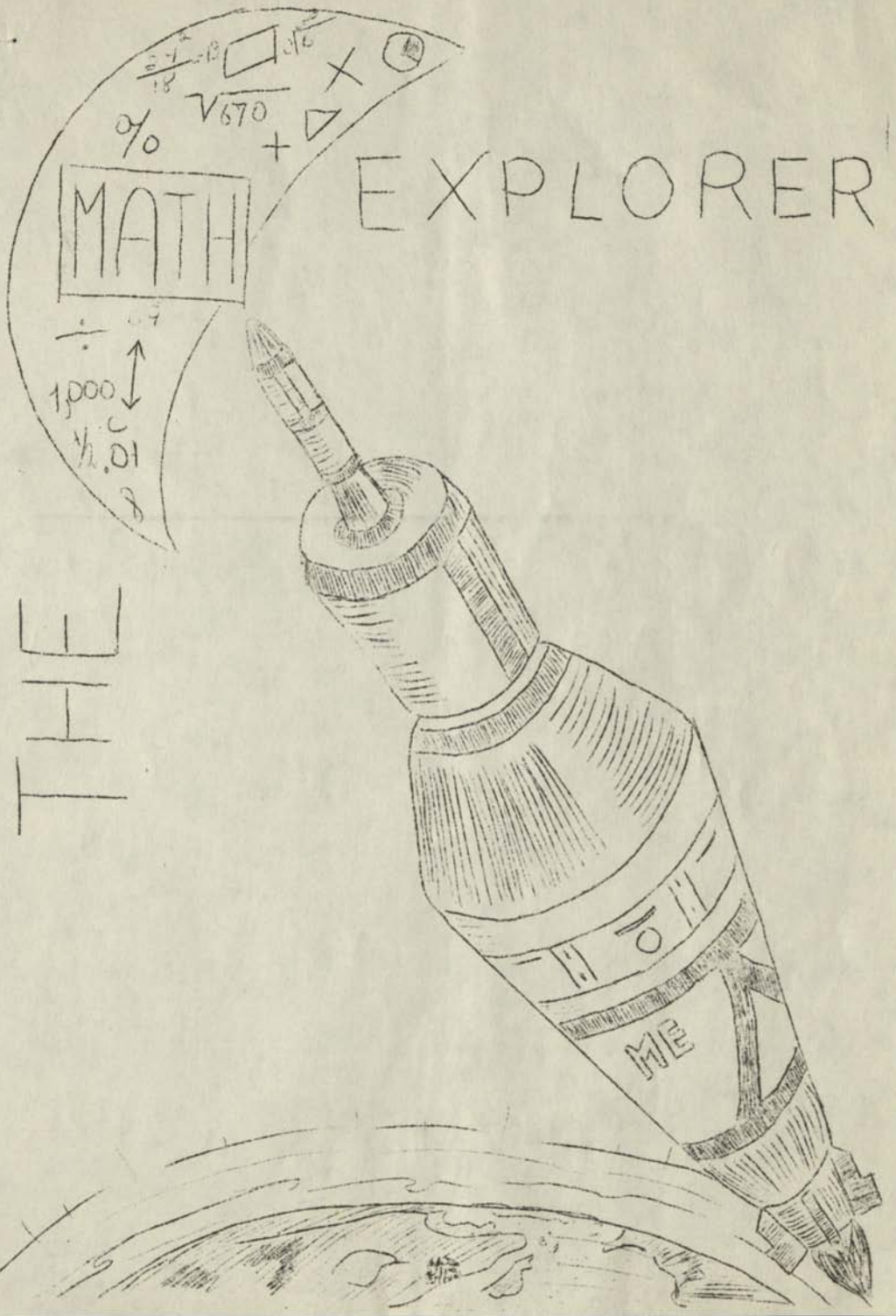
I want to thank you very much for your thoughtful letter and article for the "Math Explorer." However, it still remains a mystery to all of us on the editorial staff, as to how you received a copy of our issue in the first place. But the fact remains that you did, and we are very grateful to you for your interesting article.

I think that you brought out a very interesting point, in that, "to be truly successful, a number system must have a power of two." In our mathematics class we have been working with an experimental math curriculum that is being run, at present, by some people at Yale University. This curriculum involves studying other number bases so that we will acquire a better understanding of the decimal system. The duodecimal system was one of these. Some of us in the class wrote to the Duodecimal Society in New York, asking for material on that

system. Of course everything they sent us was written to praise the duodecimal system. I see now that it would not be such a wise choice for a number system after all.

We are very grateful to you for your interest in our paper, and would be glad to send you all further issues of this year. Naturally, we would appreciate any further comments which you would care to make.

Sincerely,
Debbie Young


MATH
%
$\sqrt{670}$
1,000
½.01
EXPLORER
THE
HOME

THE MATH EXPLORER

VOL. 1 No. 1

THE MATH EXPLORER is published monthly by the 7th GRADE MATH SOCIETY of the Weston, Conn. JUNIOR HIGH SCHOOL.

STAFF

Editor-in-Chief: Debbie Young
Assistants: Vicki Brown
George Colacicco
Steve Glazer
Sylvia Robinson
Faculty Advisor: Mr. Pierson

We are grateful to many persons for help with the mimeographing, especially Miss Maclean and Mrs. Bennett.

EDITORIAL

By Debbie Young

The purpose of this math periodical is to promote an interest in math. In it we hope to include: many interesting facts about the lives of famous or relatively unknown mathematicians; interesting mathematical ideas and principles; humorous jokes and sayings; and some math puzzles or riddles. We also hope to bring out something of the vast world of numbers - some concepts that are not taught in the everyday math class. There are hundreds of math ideas, of which many of us have no knowledge. It is these principles that we hope to explore and make clearer through the combined efforts of all who contribute to this paper.

When you think of how old the science of numbers really is, it does not seem so surprising that the numbers have gone through many stages before they became the digits we know today. The ancient Egyptians had their own numbers, and so did the Sumerians, Hebrews, Chinese, Romans, Greeks, and Arabs. Throughout their history, number systems have constantly been changing, yet remaining basically the same. Today number systems other than our decimal system are being used constantly. The binary system, consisting merely of digits zero and one, is used in electronic computers. The duodecimal system is still in the experimental stage; but there are people who believe that this system of twelve digits is more suitable to our modern life than is the decimal system. Gradually, our whole system of numbers is expanding and changing to fit our everyday lives. It is this expanding of mathematics which is always presenting new ideas, and it is these ideas, as well as interesting older concepts which we hope to make clearer to you.

MY AMBITION

By Bettina Walton

Mom, you know when I grow up,
And start out in the world,
I think I'll first be an engineer,
Or a captain with his flag
unfurled.
Or maybe even a doctor, or a
businessman or cop,
Or something else where I can soon
Begin to reach the top.

To reach my goal, I'll keep
notations,
And follow some brilliant
quotations,
Especially one said unto me,
By my math teacher, Mr. P.
Doing math is no vacation
To improve your math, you'll need
computation,
Following this quote, under this
condition,
Maybe I'll succeed in my ambition.

OCTAL

PROBLEMS FOR A RAINY DAY

By Vicki Brown

1. Georgie is a centipede with fallen arches. He has to have specially manufactured shoes made of an expensive synthetic cloth. He needs 10,000 pairs. They cost $12.50 a pair to manufacture, and were sold to Georgie at a 6% profit. What did Georgie have to pay?

2. 33 1/3% of a peanut-sheller's daily number of peanuts shelled is 50% of another sheller's output. The slower sheller shells 1,500 peanuts daily. What is the daily number of peanuts shelled?

3. Three-Hairs and Bow Legs are Indian scouts. They have to scout over 70 miles of ground. Bow-Legs can cover 3/5 mile per hour, but Three-Hairs can cover only 1/3 mile per hour. If both start from opposite ends, how many miles will Three-Hairs cover by the time they meet?

4. Gooley has a pet peacock named Irving. One day Gooley counted the number of feathers in Irving's tail. It came to a total of 64,391 feathers. Six weeks later, a stranger came along and asked Gooley how many feathers were in his peacock's tail. Gooley knew that since Irving's tail was still growing, feathers became full grown at the rate of 14 per week, and dropped out at the rate of 16 every two weeks. What did Gooley tell the stranger?

The pupils in Jimmy's kindergarten class made a list of all the words that apply to size - large, small, big, etc. Suddenly Jimmy raised his hand and said, "Oh, we forgot the most important word - 'King-size'."

THE BINARY SYSTEM - COUNTING BY TWOS

By Peter Salerno

Counting by twos, or by any number other than base ten, is quite easy if certain things are remembered which pertain to all systems. One important thing to remember is that there are only two digits in base two, 0 and 1, and all numbers in base two are composed from these two. As will be seen, even a small number will look quite large in base two, so it is impractical and hard to work with. However, the fantastic computing machines have to use this principal because they cannot work with ten digits, and they work so fast it does not matter if the number is long.

Now we must learn how the places are arranged in base two. To begin with, we know that the first place in the decimal system is one. This place has the same value in all other number systems.

The second place is always base, in this case two. In the tens system it is the tens place, in the binary system it is the twos place. So if we have 1 in the twos place and 0 in the ones place in base two, we have 1 two, or 2 in the decimal system.

Now that the pattern is shown, the rest of the place values follow it. The third place is the base times the base, so in the binary system the third place would have a value of four.

The fourth place in the binary system would be 2x2x2, or the base, times the base, times the base.

Another pattern can also be seen here. In the binary system

each place to the left doubles the value of the previous one. This is another way to find values in the binary system, but don't use it on any other system. It will not work.

Following is a chart showing the place values in the binary system, with its relation to the decimal system.

	base x base	base	ones
base ten	hundreds 100	tens 10	ones 1
base two	fours 100	twos 10	ones 1

From this chart we can see that 10_{two} means one in the twos place, or two, while in base ten it means 1 in the tens place, or ten.

Whenever you write a number in a base other than ten, you always write it with the base below and to the right of it, like this: 10_{two}. In this case you would not write it 10_2 because the numeral two might be confused with the rest of the number. Examples of numbers in other bases are: 15_{six}, 21_{twelve}.

Following is a chart showing base two numbers and their equivalents in base ten.

Base Two	Base Ten
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10

Here are some simple problems for you to try in the binary system

I. Add these numbers which are expressed in binary notation.

(a) 101_{two} + 10_{two} (b) 110_{two} + 101_{two} (c) 10110_{two} + 11011_{two}

Check by expressing the numbers in base ten and adding in the usual way.

II. Subtract these base two numbers

(2) 111_{two} − 101_{two} (b) 110_{two} − 11_{two} (c) 11001_{two} − 10110_{two}

Check by expressing the numbers in base ten and subtracting in the usual way.

EDITOR'S NOTE

We would like very much to thank George Colacicco and Sandra Hyde for our cover.

We would also like some more contributions to this periodical. There are fifty members of the Seventh Grade Math Society, but only about fifteen of these members have contributed to the "Math Explorer." We need more articles, poems, puzzles, jokes and cartoons. Now that you have seen this first issue you have an idea of what we want. Please contribute!

"Problems for a Rainy Day" will be a regular feature in the paper. We would like some more problems for this section. The answers to all this month's problems will be given in the next issue.

COMBINE WITH {MW REGION INTERVIEWS / LUMBERMANS MUTUAL 7070-800 INS CO. @ EVANSTON! } MGR APT. HARLEY THRONSON / LARRY CANTRELL, SPEC. INDUS MGR -

✓ DIST MGR

SEE JOE SWEENEY

[illegible]

CONFIRM WITH DICK KANE

~~[illegible]~~ SEC'Y MISS BURKE

Midwestern Regional Office

EVANSTON BR MGR

March 24, 1959

MEMORANDUM TO: Mr. R. W. Bemer
Programming Systems Manager
White Plains

On January 16, 1959, I wrote you regarding the possibility of your making a talk at Standard Oil Company - Indiana on the subject of Programming Systems. Attached is a copy of the letter.

Mr. (JIM) Redding is still most anxious to have you make a presentation to his group here in Chicago and for that reason, am following to see whether or not it is possible to do this within the next few weeks.

Would appreciate hearing from you very much.

10 PEOPLE RECOMM

REG. OFFICE 618 S MICH 8TH FLR

HORLIE

C. D. Lewis

C. D. Lewis

2 STL 1
3 DET
1 MINN
3 CHI 1
† CIN
KCY 1

CDL/mb
Attachment

PI 2-4800

PER GEO. FARNUM, L.I. LIGHTING IN GARDEN CITY WAS VERBALLY PROMISED, BY DATAMATIC, PROCESSORS FOR FORTRAN, AUTOCODER & FLOWMATIC

SYMPOSIUM ON THE SELECTION AND TRAINING OF PROGRAMMERS

General Practices in the U.S.A.

R. W. Bemer, Manager Programming Systems, IBM Corporation

SUMMARY

Programmers in the U.S.A. are either:

a. Tool builders (generalized programming systems which produce no answers as such), or

b. Tool users (specific applications which produce answers).

The selection of programmers thus depends primarily on which of these two functions they are to perform. A separation by manufacturer or user is artificial, since most manufacturers are actually as much customers through using computers in their own business as the normal business or scientific users. Formerly the tendency existed to separate programmers by type of application (either scientific or commercial). This distinction has been much reduced.

SELECTION

Selection of personnel is usually made from the following prime sources of applicants:

a. Personnel gravitating from other professional fields,

b. Present employees capable of being converted,

c. Personnel already trained by other companies prior in the field,

d. College-trained personnel (for only the last two years).

The procurement of adequate personnel is a serious problem in the U.S.A. For example, IBM Applied Programming is attempting to expand from 110 to 230 people in a single year. This is to produce generalized programming systems which require the most highly skilled programmers available. The selection of personnel for IBM, many of its customers, and even other manufacturers in the U.S.A. is based upon:

a. An EDPM aptitude test prepared by IBM and distributed through The Psychological Corporation, New York City. This test of reasoning power covers primarily spatial perception, inductive analogy, and problem solving.

b. Personal traits and team-ability. A lone programmer is usually quite inadequate today.

c. Previous programming experience on almost any type of computer.

d. Experience in related fields such as topology, linguistics, and information theory. (I usually try a little puzzle-solving as well.)

e. General education.

f. Prior familiarity with job to be done.

HORSE-SENSE
JACK-OF-ALL-TRADES

TRAINING

Various methods of training are employed, depending upon level of usage. These include:

a. Courses given by manufacturers on specific or general machines. (IBM alone trains over 1800 of its own programmers and over 10,000 customer programmers per year at 21 Education Centers.)

TRAINING > UNIVERSITY (110,000 OVERALL)
PUBL > PUBL HOUSE

b. Collegiate curricula. There are now 18 universities with large-scale equipment, and 83 with medium-scale equipment. Several high schools have also started successful courses.

MISSING - LOGICAL THINKING ~~THEORY~~, POLYA.

c. Manufacturers' courses for periodic retraining and enlargement of the field of knowledge.

d. Professional meetings, including three major national conferences each year, plus many by local chapters of professional societies.

e. Books and journals in this field of interest.

f. Employment of consultants for lectures and seminars by both manufacturers and users.

Another important technique used by multi-machine manufacturers is a periodic shifting of personnel to work on different computers, thus carrying along techniques in a seeding process. In such a complex field, this seeding is the most effective informal technique. The final answer to the training problem is of course to make technological advancements which eliminate the need by simplifying the job.

SDS & CONVENTIONS.

34 Ex 109 BRIT FIRMS RUN STOCK CONTROL (LEO FOR FORDS SPARE PARTS)
1/3 OF MOTOR USERS (LEO TEA STOCKS)

17 Ex 109 — INVOICING (VARYING ~~K~~, GOOD FOR COMPLICATED DISCOUNTS — WILLS IS GOOD)
(1 BAD CASE, TEMP. USED MOSTLY FOR PRINTING)

12 Ex — PRODUCTION CONTROL (~~FULLERTON GOOD EX~~, 1201, E.M.I.
L.P. FEED MIX, PETRO PRODUCT.

DIFFICULTIES 1. INSUF STAFF (NIELSENS, METRES, DID WELL, FILLED UP QUICKLY)
2. " TIME TO REARRANGE JOBS,
3. METHODS + PROGR AS = PARTNERS, BUT PROGRS WERE POOR & INEXPER.
DIFFICULT BALANCING
4. GULF BETWEEN TOP MGMT & COMPUT PEOPLE
5. SHORTAGE OF FORESEEING TYPE TO DESIGN. C.O PLACED.
SURPRISING HOW LITTLE USE OF TRAINS TIME ON COMP. — LEO HAS DONE,

TIL 2 YRS, ONLY 3 UNIV. IN FIELD.
BIRBECK APEXC (LONDON) SMITH DRUM
MANCHEST PEGR 1st, MERCURY
CAMBRIDGE EDSAC I, II — TAPES (SMALL BIG MACHINE)
PEGASUS DAGEN LEEDS, SOUTHAMPTON
DEUCE
MERC LEMO, OXFORD

FROM HUGHES, JOHN
MENAMALA UPDATER

Guide Talk: Programming Selection

I. Nature of Programmer Aptitude Test. (1953) SIMUL — RAND

A. Measure of reasoning ability.
B. Originally designed to select programming trainees (see manual).
C. Later found to correlate with programming performance.
   (a) 1956 study: correlated .356 with managers' ratings of 52 IBM 702-705 programmers.
   (b) 1958 study: correlated .439 with managers' ratings of 41 IBM 705 and 650 programmers - problem of restriction of range (average score was 59).

II. Reasons Why Test Works.

A. Reasoning ability a basic requirement for programming.
B. But other factors also important in programming - personality, motivation, etc.
C. Same situation in selection for all types of jobs.
D. Difficult to measure by paper and pencil tests - require trained psychologist to interpret.
E. Measure by usual personnel techniques - interview, application, etc.

III. 1958 Study of Personality Factors in Programming Performance.
A. Personality tests given to 41 ~~Applied Programming Department~~ IBM programmers in 705 and 650.
B. Several personality factors found related to managers' ratings of programming performance.
   (1) Original thinking (interested in new ideas, more original, more intellectual).
   (2) Ascendancy (more self-assertive, more dominant).
C. Combination of Programmer Aptitude Test plus Original Thinking correlated .568 with managers' ratings.
D. Suggests use of personality tests may improve programming selection.
E. Cautions: more cases needed, supervision by psychologist.

IV. Appeal for cooperation in study of programmers in different companies.

DR. JOHN HUGHES

Bob Bemer

No. of Programmers Trained - 1958

		IBM		Custom
305 -	Basic Sales	678		-
	Programming	870	870	2353
		1,548		2,353
650 -	Basic Sales	747		-
	Programming	428	428	5,045
		1,175		5,045
704		140		457
705		333	558	2,107
709		85	1856	378
		558		2,941
	GRAND TOTAL	3,301		10,364

Not included are 7070 users which are in Englewood operation and not true programming schools. Also not included are those schools which offer IBM programming but are legendarily

Application Form

Institute on Digital Computer Applications

June 2, 3, 1959

Kindly reserve places for the following persons:

	Name	Mailing Address	Company
1.			
2.			
3.			

Check(s) in the amount of $_____ attached (fee of $20 per person) payable to Marquette University.

Mail to Raymond J. Kipp, Director of Engineering Institutes, Marquette University, 1515 W. Wisconsin Avenue, Milwaukee 3, Wisconsin.

Separate along this line

Retain program information for your files.

Marquette University College of Engineering

INSTITUTE ON DIGITAL COMPUTER APPLICATIONS

June 2, 1959

8:30 a.m. Late Registration

9:00 a.m. Welcome: Dr. A. Bernard Drought
Dean, College of Engineering

9:10 a.m. Presiding: Dr. James Horgan, Director
Department of Electrical Engineering

"MODERN NUMERICAL METHODS IN DIGITAL COMPUTATION"
Prof. Robert E. Frounfelker,
Marquette University

10:15 a.m. Coffee Break

10:30 a.m. **"MODERN NUMERICAL METHODS IN DIGITAL COMPUTATION"**
Prof. Robert E. Frounfelker

12 Noon Luncheon—Room 101

2:00 p.m. **"INTERNATIONAL ALGEBRAIC LANGUAGE"**
Dr. Robert Bemer, Manager
Programming System, IBM
White Plains, New York

3:00 p.m. Break

3:15 p.m. **"APPLICATIONS OF DIGITAL COMPUTERS IN ELECTRIC POWER INDUSTRY"**
Dr. Tomas Glass
Marquette University

4:30 p.m. Adjourn

ED BATTISTE
CHUCK JOHNSON

June 3, 1959

Presiding: Prof. Stanley Krupnik
Department of Electrical Engineering

9:00 a.m. **"APPLICATION OF DIGITAL COMPUTERS IN PROCESS CONTROL"**
Dr. R. Curtis Johnson
Process Applications Department
The Thompson-Ramo-Woolridge Products Co., Los Angeles, California

10:15 a.m. Coffee Break

10:30 a.m. **"APPLICATION OF DIGITAL COMPUTER IN DESIGN"**
Dr. Eldo C. Koenig
Engineer in Charge
Engineering Analysis Section
Allis Chalmers Manufacturing Co.

12 Noon Lunch

2:00 p.m. **"DIGITAL COMPUTERS AS A RESEARCH TOOL"**
Mr. Richard Haertle
Engineering Math Group
A. C. Spark Plug Division

3:00 p.m. Break

3:15 p.m. **"NEW DEVELOPMENTS IN DIGITAL COMPUTERS"**
Mr. Edward L. Battiste
Applied Science Representative
I.B.M., Milwaukee

4:30 p.m. Institute Closes.

General Information

The rapidly increasing use of digital computers in solving scientific and engineering problems has made it imperative that the well versed engineer be familiar with this modern tool.

The general purpose digital computer offers increased efficiency of engineering manpower and is many times the only practical way of solving a given problem.

This institute is arranged to acquaint the practicing engineer with various aspects of the field of digital computaton. The sessions begin with modern numerical methods followed by the programming of digital computers, progress through various fields of applications, and end with a look at the future. Audience participation in question-and-answer periods is invited.

Our Golden Anniversary

50 YEARS OF SERVICE

College of Engineering

The

Department of Electrical Engineering

of

MARQUETTE UNIVERSITY

announces

AN INSTITUTE ON

DIGITAL COMPUTER APPLICATION

June 2, 3, 1959

Room 206

BROOKS MEMORIAL UNION

620 North 14th Street

Milwaukee 3, Wisconsin

1907 - College of Engineering - 1958

DRAFT-Mr. D. R. Wright
May 12, 1959

The 701 and 702 marked IBM's entry in the field of stored program computers. The stored program concept is that single feature which distinguishes the modern computer from the calculating tool and allows the computer to perform virtually any logical or arithmetic process. The secret is simple. The machines' own instructions are stored in its memory in the same form as the numbers upon which it operates. This sequence of instructions is called a program; it controls what the computer is to do. Since this program is stored as numbers, it is possible for one part of the program to perform arithmetic operations upon another part of the program not then in use. After this, control of the machine may pass to those instructions which have been modified. These will then be performed in this altered form. Though the physical means of representing, storing, and operating upon data have changed greatly in the eight years since the 701 and 702 were produced commercially, this stored program concept remains virtually unchanged. It is this that gives us the power to translate from one language into another or from the language of the computer user to the simple numbers that the machine understands as instructions.

The 701 was basically for scientific use and all its numbers and instructions were represented in the binary system, that is, it understands internally only the digits 0 and 1. In contrast, the 702 was primarily a commercial machine for accounting and other forms of data processing. It handled not only the digits 0 through 9 but could also represent the letters of the alphabet and many other special characters internally.

What we call the memory of the computer is very interesting. It is usually some physical device that we use to store a great many numbers for instructions or data. The unique feature is that the computer is able to erase old data and replace it by new. Consider a magnetic tape recorder. The same record can be played many times but when you record something new the old pattern is erased and the new takes its place. Magnetic tape just like this is one of the major means of storing data in computers, particularly for large amounts of information. Compared to other types of storage, magnetic tape is relatively inexpensive but has the drawback that it must be searched serially. This is as though you were looking for a book in a library and always had to search for the book in the same pattern, going along each shelf until you find it.

In contrast to this there are types of memories known as random access. This corresponds to being able to go directly to the book you want because you know beforehand they are ordered on author. To find something by "Jones" you look for something in the "J" section, follow to the "Jo", "Jon," etc. Each element of a random access memory has a number associated with it which is an address just like the address on your mailbox. This is a far more expensive type of memory than magnetic tape but it operates much faster.

The random access memory of the 701 and 702 was a series of cathode ray tubes similar to a television tube. Later, these memories were changed to be made up of thousands of tiny magnetic donut-shaped cores. With other technical improvements these new machines were then designated as the 704 (scientific) and 705 (commercial and business).

Another type of memory which compromises between magnetic tape and magnetic cores is the magnetic drum. The elements of a drum also have addresses but each one is identified in two ways, how far along the drum and how far around the drum. These magnetic drums were available as auxiliary storage on all of the computers mentioned here but they were also used as a basis for a smaller and cheaper computer, the IBM 650. Hundreds of these computers have been produced. They were particularly popular because of the self-checking built into the machine. This makes it virtually impossible for an error in arithmetic to occur without being indicated to the user. Reliability of this type is particularly vital to the commercial user.

Still another type of memory has been developed, known as RAMAC, which is in effect like a stack of phonograph records. Each item of information is found by the number and side of the disk, the number of the track, and the position along the track. These tracks might be said to operate much as the grooves in a phonograph record do. Being able to find the place of information by four dimensions like this in a not too expensive medium opened up vast possibilities of accounting control, particularly for invoicing, inventory, etc. Such a single disk file may contain five or ten million digits or letters.

Many components of these machines still contain patch panels for wiring (interconnection like a telephone switchboard), but the basic mode of control is still the instructions put into the memory of the machine itself. Since these sets of instructions may be entered in any form or replaced, these are said to be general purpose computers.

They can be controlled to solve virtually any problem. A special purpose computer, on the other hand, would be present at the factory to do only a single job. Thus, it knows what job it is going to perform before it is delivered. However, the general purpose computer is delivered to you in complete ignorance of the job you intend to use it for. You must create the instructions that make it do your job. This is known as programming.

Basically, programming consists of creating a series of numbers which are read by the machine as calling for certain operations. Thus, a 3 in the first digit of a number might call for an addition, 4 in the first digit might call for multiplication. This is called machine language. It soon developed that communicating to the machine in its own language was very tedious, time-consuming, and expensive. Extensive rules made it difficult to correct and change programs. We found ourselves in the difficult position of having to talk to a super-fast machine in very child-like language. In short, our robot was uneducated. Since we said the machine can do any job, one of the things that it ought to be able to do is translate from our language into its own, and this is the way computers are operated now days. The process is known as automatic programming. Programming systems such as Autocoder and FORTRAN are synthetic languages, created by IBM, which the user of the machine can use to state his problem and through them the machine can translate his statements into its language. In FORTRAN, for example, you need not write anything except an algebraic expression. The computer will figure out how it should be solved. Thus, every problem takes on a two-part aspect. The first is where the computer uses itself to translate from your language to its language; the second is where it takes these results to obey these instructions to produce answers. This may at first seem like a wasteful process, but actually humans are too inefficient to write instructions fast enough to keep up with the machine. If we had not already had such systems we could never have gotten the 704 in the Vanguard Center at Washington ready to track satellites.

In addition, these automatic programming systems have opened up high-speed computers to a much larger class of users. Formerly only an expert could make the machine do what he wanted. Now firms such as North American Aviation train all their engineers in FORTRAN so each one can take advantage of the vast power of the computer. In two years North American has trained over 1000 engineers to use these machines simply. In short, we have educated our robot so he can understand mathematics in the language that we use.

A significant development in the use of computers has been the formation of user groups such as SHARE and GUIDE. In order to do more and more complex work on a computer the vast amount of efforts required even in these synthetic languages make the individual user bog down after he has gone so far. Here the stored programs show a curious feature. It cost almost as much in Detroit to build the ten thousandth automobile as it does the twentieth. But once a program has been written it can be run on all computers of one type. The only cost is a few dollars for copying the punched cards which contain the program. SHARE and GUIDE are cooperative groups. Each participating installation and company does a small part of the job required to tool up a computer. Even though their eventual applications may be different, there is much basic work in common between a company using a computer to design airplanes and another using it to do medical diagnosis.

This altruistic sharing of effort by public-minded users has been the greatest impetus in the effective and vast usage that the U.S.A. has been able to accomplish. Continued development may be the most significant social force in our generation. It is well known that a few years ago Russia ranked the development of electronic computers as their number one priority, even ahead of atomic energy. IBM can be proud of being a leader in the movement which does not, as once was thought, replace the human with a machine, but rather releases him from the drudgery of tedious and repetitive work so that his full creative capabilities may be realized to the advancement of society.

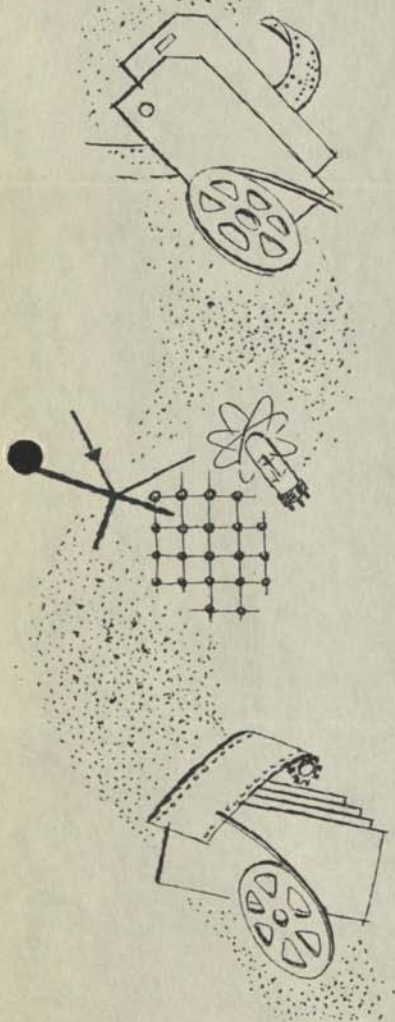
rwb/cac

About the Authors . . .

William Thayer, author of the article ". . . New Designs Needed for 1,000° F" (see page 6) received a B.E.E. degree from Cornell University in 1951. Following graduation, he did instrumentation development work for the Flight Test Department at Boeing, Wichita. Later he joined the staff at Cornell Aeronautical Laboratories, Inc., where he became a Project Engineer involved with artificial stability and control systems for both aircraft and experimental automobiles. In 1957 Mr. Thayer moved to the Moog Valve Company, Inc., where he became Chief Development Engineer. Just recently he was made Assistant Chief Engineer.

R. W. Bemer, ("Evaluating Intelligence for Programming Systems", p. 22DC) a mathematician and engineer who is well known nationally within the computer field, is associated with IBM in the capacity of Manager of Programming systems. He received a B.A. in Mathematics from Albion College in 1940. Later, at Marquardt Aircraft Company, he created the Numerical Analysis Section and the computing installation. Similarly at the Lockheed Aircraft Missile Systems Division, he was responsible for the forming of the Mathematical Analysis Section and the computing installation, both digital and analogue. Mr. Bemer has worked on Friden 602, 602A, 604, 607, CPC. He has programmed for E101, 650, DATATRON, CADAC, SWAC, and many others . . . and has directed development in automatic coding on FLAIR (650), PRINT I (705), FOR TRANSIT (650), XTRAN (705, 709).

Announcing



A SYMPOSIUM ON

RECENT ADVANCES IN

PROGRAMMING METHODS

Saturday

March 21, 1959

The Ohio State University

Columbus, Ohio

SPEAKERS:

Edward W. Cannon
National Bureau of Standards
Washington, D. C.

Robert Wm. Bemer
International Business Machine Corporation
White Plains, New York

Walter A. Ramshaw
United Aircraft Corporation
East Hartford, Connecticut

William C. McGee
General Electric Company
Richland, Washington

John Matousek
Systems Development Corporation
Santa Monica, California

Jules Mersel
Space Technology Laboratories
Los Angeles, California

Conducted by

THE CENTRAL OHIO ASSOCIATION FOR COMPUTING MACHINERY

Program & Registration Information will be furnished at a later date

Jack Belzer, Chairman
Battelle Memorial Institute
Columbus 1, Ohio

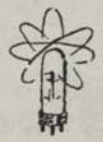
NOTES

A SYMPOSIUM ON RECENT ADVANCES IN PROGRAMMING METHODS

Saturday March 21, 1959

The Ohio State University

Columbus, Ohio



Conducted by

THE CENTRAL OHIO ASSOCIATION FOR COMPUTING MACHINERY

Jack Belzer, General Chairman
Battelle Memorial Institute
Columbus 1, Ohio

MORNING SESSION
Hagerty Hall

Chairman: Fletcher Jones, North American Aviation

9:00 "Greetings"
Professor Roy Reeves
President, C.O.A.C.M.

9:05 "Welcome"
Gordon Carson
Vice President, The Ohio State University

9:15 "Designing a Programming System for Human Convenience"
Robert Bemer
I.B.M. Corporation

Intermission

10:15 "Using the Here-to-Stay Computer as an Engineering Tool"
Walt Ramshaw
United Aircraft Corporation

Intermission

11:15 "A Cooperative Venture in Generalized Systems Programming"
William C. McGee
General Electric Company

LUNCHEON SESSION
New Orleans Room
Jai Lai Cafe

12:30 Luncheon

1:30 "Evaluation of Programming Methods"
E. W. Cannon
National Bureau of Standards

AFTERNOON SESSION
Hagerty Hall

Chariman: Robert Kissinger, Nationwide Insurance Company

3:00 "The Professional Programmer's Automatic Programming"
Jules Mersel
Space Technology Laboratories

Intermission

4:00 "Special Programming Techniques in the Development of Large Scale Computer Programs"
John Matousek
Systems Development Corporation

Advance Registration Form Enclosed

DESIGNING A PS FOR HUMAN CONVENIENCE YS EDUCATES THE PROG

THUS, A FACILITATED MAN-MACHINE RELATIONSHIP

1. COMMUNICATING
   a. LANGUAGES RIGOR, NOT STUTTERING
      SYMBOLS — 4000 INTERLINGUA (2^{12}), INFO THEORY & CONTENT, BRAILLE (6)
      " MANDARIN, 48/64 NOW - FUTURE?
      IAL - NEW HOLLERITH → 256/512 ($2^8/2^9$) MATHEMATICS NEEDS MANY FOR CONCISENESS, CLARITY & QUICK ANALYSIS. (BIN/OCTAL/DEC # SYMBOLS)
      PUNCTUATION & MAX. MEANING.

2. FLOWCHARTS & ORGANIZATION
   HUMAN CAPACITY TO ORGANIZE INTERNALLY. VISUAL □ SPAN. TITLING FOR INDEXING

3. TRANSMISSION — INPUT
   VARIOUS MEDIA, SPEED VS. COST (UNIT × USAGE). 2) TYPES DATA & INSTRUCTIONS — REQUIRE DIFF TREATMENT. OFF-LINE VS ON-LINE. COLLAT PROBLEMS OF EPCH VS DIRECT TYPE, NEED TO HUMAN CONVENIENCE.

4. TRANSMISSION — OUTPUT
   2 TYPES, PROG OUTPUT & ANSWERS. ANSWERS STFWD WITH FORMAT, BUT PROG REQUIRES:
   a. UPDATED RECORDS — RE PROCESSING
   b. BACKTALK
   c. LOGGING
   d. PARTIAL DIAGNOSTICS + GUESSES.
   e. RETURN INFO IN SAME LANGUAGE.

5. DIAGNOSTICS
   MEMPRINT (BARE BONES + CHARS BLKS, DOT. DEFGMT)
   ORGANIZ FOR MINIMUM DIAG (AUTO FLOW CHRT MAINTENANCE)
   TESTING = FLOW BY PROPER PLANNING.

COACM

21 MAR 1959

II Mode of usage

PERM LIBRARY / TEMP — DO SAME AS LAST WK, EXCEPT—
JARGONS AS SUPERSTRUCTURES IN HIERARCHIES

MIXED PROD & TESTING ∴ SUPERVISORS FOR HSEKG.

EXECUTIVE CONTROL, (SIMILE) & OVERRIDE FOR PRIORITY & THINGS THAT ARE TOO EXPENSIVE TO PROGRAM,

III PLANNING & LOGISTICS OF PROCESSORS

Processing vs. running. Inordinate balance not allowed. Reprocessing is the normal mode. How to save logistically by symbol manipulation and flow algebras. Show 709 IAL schema. Do not los3 information once gained. Modular processors and detour of unneeded elements, Sortigga sthe most powerful internal tool. AC macro lookup example. Double refenerce.

Cross searching among programs by differnet programmers. How learner program can combine and improve, super IQ.

Learning, reordering and scrapping old systems. Abililty to simulate processor elements in different orders or mode of operation.to find optimum (cheap optimum)

DPDHQ, White Plains
December 22, 1958

Memorandum to Mr. R. W. Bemer

Subject: Programming Conference at Carnegie Tech
February ~~5 and 6,~~ 19 20 1959

This will confirm our discussions on this subject. You are going to participate as IBM's official representative. To really gild the lily you will pay the $25 registration fee if they will accept it.

I appreciate your willingness to help us resolve this rather embarrassing situation.

M. A. Shader

MAS/bp

cc: Mr. R. W. DeSio - Midwestern Region
Mr. A. L. Harmon - DPDHQ

DON KNUTH

Midwestern Regional Office

January 13, 1959

Memorandum To: Mr. R. Bemer

Subject: Carnegie Rand Information Processing Language

It would seem that the 7070 would be ideal for writing a subject language code.

If this is correct and if you have the time and interest to formulate some supporting suggestions, I would be pleased to arrange to have you meet with Simon and Newell to discuss the same during your visit to Pittsburgh next February 19th and 20th (to attend the Michigan, Case, Carnegie 650 Users Conference).

Such assistance; if possible, could have a very favorable effect on present 7070-220 deliberations at Carnegie.

Thank you, Bob.

Bob

R. W. DeSio

RWD:jg

John Lawrence - CHI IBM
U Rep

SPEED > JOHNIAC
DISK CHANNING / BUF IN BLOCKS
800 MO MAX
153-300 MIN

INTERP.

— 99 INDEX WORDS/REG
— FIELD DEFINITION (MASK)
TLU (GOAL) (LOWEST) ~~(HIGH)~~
— 30 SWITCHES (0101 - 0103)
BRANCH & LOAD INDEX (RU)
— SCATTER R-W

KANSAS CITY CHAPTER ACM

PROGRAM FOR 1958-1959

November 1958

R. W. Hamming, Bell Telephone Laboratories
"The Future of Computing"

December 1958

Charles Katz, Remington Rand UNIVAC
"Automatic Programming"

January 1959

J. F. McMurrer, System Development Corporation
"SAGE"

12 February 1959

R. E. Bemer, International Business Machines
INTELLIGENCE IN PROGRAMMING SYSTEMS

March 1959

Dr. Walter Bauer, Space Technology Laboratories
"Computer System Design"

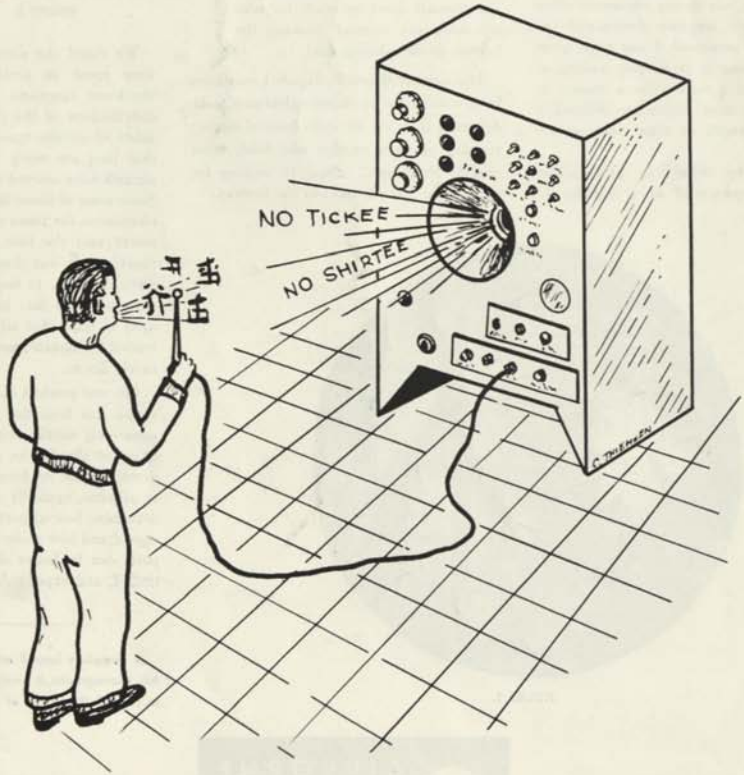
April 1959

Don Madden, System Development Corporation
"Wages and Salary in the Computing Field"

Vol. 7, No. 6
February, 1959

The PULSE of

LONG ISLAND



SECTION MEETING FEBRUARY 10

"MACHINE LANGUAGE TRANSLATION"

ROBERT W. BEMER

STRATFORD AVENUE SCHOOL

Air traffic, a subject which has long been a matter of concern at Airborne, is further discussed this month with particular respect to landing strips.

AIRPORTS

We have been looking at airports lately. In our concern with the state of air traffic we have turned our eyes downward from the airways and radio fixes to those apparently simple strips of concrete that are the origins and destinations of all that traffic.

We have been peering at those airports with the naked eye, with high-powered binoculars, and through the electronic eyes of the ASDE radar of which we are so proud. Usually, our timing clocks are close by as we watch; we have determined (as you may have suspected if you have spent much time sitting in an airliner waiting at the entrance of a runway for a chance to take off) that most flights are delayed a considerable length of time right at the airport.

Much of the delay for a departing aircraft is experienced in a portion of its journey, which we have termed the queue. The queue is the line-up of eager aircraft, engines racing or idling, that is formed at the runway entrance. We have counted as many as ten aircraft in the queue on a fairly normal day at Idlewild. Figure 1 shows a queue as observed on the ASDE. In Figure 1 eight aircraft can be seen waiting along the taxiway (at the top of the photograph) and one aircraft lined up ready for take off down the runway (toward the bottom of the photograph).

The aircraft shown in Figure 1 may have been waiting for an enroute clearance to be delivered from the air traffic control center; running up their engines and doing other cockpit check-outs, or merely waiting for an arriving aircraft to clear the runway.

FIGURE 1.

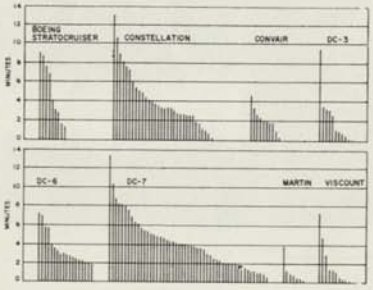


FIGURE 2.

We timed the aircraft to determine the time spent in performing the required check-out functions. Figure 2 shows the distributions of the times required for the pilots of various types of aircraft to report that they are ready for take-off after the aircraft have entered the queue at Idlewild. Since none of these flights required enroute clearances, the times shown in Figure 2 represent only the time required for cockpit functions. It was disappointing, to the analytical minds, to find that these distributions were not flat; however, we have managed to learn that aircraft are usually controlled by humans, and human reactions are rarely linear.

The end product of all this airport observation has been the realization that these apparently simple strips of concrete are not quite so simple. As a result we have the determination to devote more time looking at airports, optically and electronically, to determine how airports might be better designed and how traffic on good and bad airports can be better observed, guided, controlled, and expedited.

A complete bound set of our third series of AIL Monographs is available on request. Write to Harold Hechtman at AIL for your set.

LAST IRE - PGEE CALIF - KICK IT.

THE COMPUTER - STARTING POINT FOR A NEW ERA

WHAT DOES IT DO. TURING G.P. - UNIVERSALITY.

HOW OPERATE - SYMBOLS, CHARACTERS, EXP CHAR SET FOR FUTURE USAGE = 256
NEW BREAKTHRU.

SIMUL TRANSMISSION, LINOTYPY,

SYMBOLS → LANGUAGES, NOT ALL SPOKEN (INFLEXION, ROBOT - ASF - NO? QUESTION, SARCASTIC)

LANGUAGES - ORDERED SEQ, SCANNING - IN LINE (SUPRA SUB)

RECURSIVE PROPERTIES (PAPER, EXAMPLES MIRROR MAG LENS)
└ INTERLINGUA, ESPER, RECURSION MINIMIZES GLOSSARY

DIGITAL VS ANALOG, WHATUM SAID.
SEE CARD 4-PBL-Δ!

TRANSFORM TO M.L. TALK TO MACHINE

NOW

1. PROBLEM LANGUAGES
   FORTRAN, IAL, COMTRAN, FLOWMATIC
   SYSTEMS CONCEPTS
   INTELLIGENCE
   FEEDBACK

IRE - L.I.
10 FEB '59

2. COMPUTER LANG. (INTERCOM UNCOL)

DATA HANDLING & DESCRIPTION XXX, XX, ALL AUTO (BENSIN-ALBUQUERQUE)
SPECIAL DEVICES — ML TO ML TRANS IMPOSS.

2. MECH TRANSLATION -
   LIST WORKERS, CURRENT SUCCESS HISTORY - GEORGTOWN, 701 EXCEPTION BASIS
   LINGUISTIC VS LOGISTIC (SORTING, PASS LIB. ONCE, SIGN LING 3 GROUP)
   HOWE SPEED, NEW DEVICES (RAND MEMORY) OUT OF SIGHT, OUT OF MIND, BLIND INDIA
   RUSSIAN WORK VS OURS m! VERSUS ZM STAR ≡ UNCOL
   NEED FOR SPECIAL TOOLS, TABLE OPS, SPEC LANG - US!

3. INFO RETRIEVAL
   LIST WORKERS
   IBM WORK - LUHN (AUTOABSTR) RECURSION AGAIN - INFO², PERLIS STORY
   └ PAMPHLETS, NEW ORG TO - BATCH) RES.
   MEMORIES - RAMAC, PHOTO

4 PERCEPTRON, LEARNING, SELF-ADJUST TO ENVIRONMENT, WRITE MUSIC, etc.

Machine Language Translation

Section Meeting

STRATFORD SCHOOL — FEBRUARY 10

MURRAY NOVICK

"It's Greek to me" is an expression used by most people (other than Greeks) to indicate that they do not understand. With machine language translation just around the corner, however, this saying may soon go to the way of the buggy whip.

Machine translation by means of computers will be one of the topics discussed by Robert W. Bemer, guest speaker at the next meeting of the L. I. Section. The meeting will be held on Tuesday, February 10, in the Stratford School, Garden City.

Machine language translation is possible because of the computer's ability to perform logical manipulation of strings of symbol representations. It is this property that will provide the key to duplication or simulate the actions of living organisms. In addition to machine language translation, other interesting fields where work of this nature is being done now are abstracting and information retrieval, and the creation of synthetic computer languages for man-machine communication in the solution of problems.

Mr. Bemer will survey present work and approaches in these allied fields and speculate on future capabilities.

Robert W. Bemer is Manager of Programming Systems at IBM. Previously he was supervisor Numerical and installation, both digital and Analysis Section (created department and installation digital and analog) at the Marquardt Aircraft Corporation, California. Before that, he was group leader, Mathematical Analysis, at the Lockheed Aircraft Corporation.

He received his A.B. in mathematics from Albian College, Michigan, in 1940.

Mr. Bemer is also Editor of the Techniques Department of the "Communications of the ACM."

PRE-MEETING DINNER

A pre-meeting dinner will be held at 5:45 P.M. at Howard Johnson's Restaurant on Jericho Turnpike in Mineola, N.Y. This will be an excellent opportunity to meet the speaker and discuss the topic of the evening.

ROBERT W. BEMER

COVER:

It's amazing what a machine can do ! !

THE COMPUTER AS A SYMBOL MANIPULATOR

One of the most fascinating properties of a general-purpose computer is the ability to perform logical manipulation of strings of symbol representations. It is this property that will provide the key to duplicate or simulate the actions of living organisms. Among the most interesting fields where work of this nature is being done now are the three allied subjects of mechanical language translation, abstracting and information retrieval, and the creation of synthetic computer languages for man-machine communication in the solution of problems. Present work and approaches will be surveyed, together with some speculation on future capabilities.

EDITOR'S NOTES

A new constitution for the I.R.E. has recently been placed before the membership for their approval. This constitution is the result of four years of study dedicated towards producing a set of rules which will enable the I.R.E. to cope more realistically with present day requirements.

It is necessary for at least twenty percent of the total voting members to send in a ballot in order to effect a change in the constitution. This is a wonderful opportunity for our readers to demonstrate the vitality of our Long Island Section by registering a 100% vote. Let us show that those of us living in Long Island, the cradle of electronics, are interested in the functioning and administration of our professional society.

• • •

AIEE CONVENTION

A panel discussion entitled "Communications in Space" will be held at the Hotel Statler on February 4, 1959. This after-lunch session will begin at 2:30 P.M.

The five panel speakers are well known to Long Island Section audiences. Dr. Pierce, of the Bell Telephone Laboratories, and Dr. Vogelman, of the Rome Air Development Center, have recently spoken at the L. I. Section meetings. Completing the group are Prof. Fano of MIT, Dr. Shuey of General Electric, and Major Gillen of ARDC.

As Dr. Hellmann pointed out in the chairman's article the desireability of cooperation with the AIEE as to membership requirements is one in which the initiative is to be by the IRE. In this respect the one day registration fee for this event of $2.00 to both members and non-members is a new experiment. The registration for the whole week's events, on the other hand, is $5.00 for members and $8.00 for non-members.

The PULSE
of Long Island

Published by the
LONG ISLAND SECTION
Institute of Radio Engineers
February 1959
Vol. 7 No. 6

Published monthly, except July and August. Editorial and advertising deadlines are the 25th of the second month preceding the month of issue. Address all correspondence to:

THE PULSE OF LONG ISLAND
Long Island Section, IRE
P. O. Box 789
Mineola, New York

Second-class postage paid at Baldwin, N.Y. Office of publication located at 87 Merrick Road, Baldwin, N.Y. The PULSE is mailed to members of the IRE who give as their address a Post Office in Nassau, Queens, or Suffolk Counties. Subscription fee of $1.00 included in IRE membership.

Notes - WHAT HAPPENED TO COMPILERS , Milwaukeev ACM, 5 Feb 59

1. Multi-programming

3 fine examples on TRC 705, but in general with this class of equipment you might find that timing tolerances will not accept much change. Too delicately balanced.

Proper shared operation of a single computer depends on having each component being able to signal a priority control station at the beginning or end of its recognized process. This priority table must be available to a decision program of considerable complexity and intelligence.

ACM, Milwaukee Chapter, 5 Feb 59
WHAT HAPPENED TO COMPILERS?
New IBM computers, by their multi-programming features and simultaneous read-write-compute, are obsoleting the compiler principle that says you know how a program should be constructed before running it. In COMTRAN, for instance, the I-O supervisor will operate interpretively to adjust the program to variations in data flow. Other processors have seldom considered this factor and we believe this approach is worthy of considerable emphasis. (STRETCH data-word limited indexing, Trapping, use share memory with config. memo, self-adjusting systems to configuration, possible machines with indirect addressing etc. to act simultaneous so nterpreters will operate equivalently fast. Stretch look-ahead. Compile on demand, or generate.)

further instruction. Among the items in this inviolate area, created by the Macro FENCE in Autododer (statically), is a list of all possible machineunits, with indicators for both ready and online status. CT fills in this information and file protects the area. If the equipment shown is sufficient to handle the processor, the program proceeeds. If not, a printer message informs the operator. When machine units are represented symbolially in the source program, the tables of correspondence are built up at this time. This is another item that cannot be compiled in advance.

The processor may be self-adjusting in size and logic according to the equipment it finds available, with certain portions generated on the spot just before running. ONLINE GENERATION PRECLUDES COMPILING.

A pseudo-operating CONFIG may exist inthe source program to be processed. This may specify the minimum machine allowable to run the object program. If the processor encounters this, it copies CT from itslef into the object program so that it will be the first executed section, which means that no time willbe wasted in trying to runthe object program on less than the required machine. Another pseudo-op INTERR in the source program gives the time interval for each interrogation of a start signal fromthe equipment used in semi-online fashii n. This creats the master-slave type of operation rather than a priority interrupt. Either the principal job (using the most equipment) or a supervisory program is the master. All other usage slaves to this. The master asks, at stated intervals, DOES ANYBODY WANT TO GET ON. If the answer is NO the processes proceed. If yes, control transfers to a portion of

Notes - WHAT HAPPENED TO COMPILERS , Milwaukeev ACM, 5 Feb 59

1. Multi-programming

3 fine examples on TRC 705, but in general with this class of equipment you might find that timing tolerances will not accept much change. Too delicately balanced.

Proper shared operation of a single computer depends on having each component being able to signal a priority control station at the beginning or end of its recognized process. This priority table must be available to a decision program of considerable complexity and intelligence.

2. Shared Online Operation

All hardware must be completely programmed controlled, and have a programmable clock with a program interrupt feature. The basic program unit might be called CT, a configaration tester. A initial version of this is being placed in Autocoder III.

CT exists in every processor, such as a translator from synthetic to machine language or a sorting generator. It is one of the first units to operate in the processor and interrogates all possible equipment to determine what is on line and in ready status. An inviolate area of memory is reserved for control. Read-only memory would be desirable, or it would be preferable that this area of memory be settable by hardware instructions such that there is a programmed fiel-protect over a specified blockof addresses untilreleased by a further instruction. Among the items in this inviolate area, created by the Macro FENCE in Autododer (statically), is a list of all possible machineunits, with indicators for both ready and online status. CT fills in this information and file protects the area. If the equipment shown is sufficient to handle the processor, the program proceeeds. If not, a printer message informs the operator. When machine units are represented symbolially in the source program, the tables of correspondence are built up at this time. This is another item that cannot be compiled in advance.

The processor may be self-adjusting in size and logic according to the equipment it finds available, with certain portions generated on the spot just before running. ONLINE GENERATION PRECLUDES COMPILING.

A pseudo-operating CONFIG may exist inthe source program to be processed. This may specify the minimum machine allowable to run the object program. If the processor encounters this, it copies CT from itslef into the object program so that it will be the first executed section, which means that no time willbe wasted in trying to runthe object program on less than the required machine. Another pseudo-op INTERR in the source program gives the time interval for each interrogation of a start signal fromthe equipment used in semi-online fashii n. This creats the master-slave type of operation rather than a priority interrupt. Either the principal job (using the most equipment) or a supervisory program is the master. All other usage slaves to this. The master asks, at stated intervals, DOES ANYBODY WANT TO GET ON. If the answer is NO the processes proceed. If yes, control transfers to a portion of

International Systems Meeting

October 13-14-15, 1958 • Pittsburgh, Pennsylvania

W. N. SPRAY
General Chairman

December 17, 1958

Mr. R. W. Bemer
International Business Machines
590 Madison Avenue
New York 22, New York

Dear Mr. Bemer:

I wish to express the sincere appreciation of the ISM Committee as well as my personal thanks to you for your cooperation in making our "professional exhibits" at the 1958 International Systems Meeting a success. Will you kindly express our appreciation to Mr. D. A. Hemmes.

As you may know, due to a hotel strike in Pittsburgh, it was necessary to move the Convention to Buffalo, New York. While the details involved in making this move were numerous, the project was successfully completed.

Your manuscript entitled, "Computer Language Compatibility Through Multi-Level Processors", is being returned herewith. Upon receipt of same, will you kindly mail us the enclosed card so that we may be assured of its safe arrival. No manuscripts are being published in the proceedings of the meeting.

It may interest you to know that about 900 delegates attended the ISM and the Systems Panorama which were jointly conducted.

Our "reading room" in which your manuscript was displayed was one of the busiest exhibits at the meeting.

Again, allow me to express my personal thanks for your help and for your generosity in loaning us material for display.

Very truly yours,

R. P. Vaughan

R. P. Vaughan, Member
Professional Exhibits Committee

ajo

Attachment

THE UNIVERSITY OF CHICAGO
CHICAGO 37 • ILLINOIS
THE SCHOOL OF BUSINESS

January 16, 1959

Dr. R. W. Bemer
IBM Corporation, Data Processing
Division
112 East Post Road
White Plains, New York

Dear Dr. Bemer:

Please excuse my tardiness in conveying our thanks to you for speaking before the Mid-Continent Computer Club in December. As you could see, your presence brought out a goodly number of members, all of whom seemed very interested in your subject.

Thank you also for the material you sent to me personally; I'm looking forward to perusing the International Language Specifications more thoroughly when the Communications appears.

Again, let me express the appreciation of the Club for your visit.

Sincerely yours,

Robert L. Ashenhurst

Robert L. Ashenhurst
President, Mid-Continent
Computer Club

RLA:uw

"HOW SMART ARE YOUR PROGRAMS" Chicago, Mid-Continent Computer Club (MC_2)
5 Dec 1958

2 classes of programs 1. General - no answers
2. Applications

In all, USE THAT MACHINE TO PRODUCE USEFUL WORK, not optimum holding (Oscar)

GENERAL - USER

Opdrating characteristics, system specs, then language specs
How it should operate for the user-programmer
Source map into object, fast reassembly, FORTRAN to Russiaas, wrong tack - reaction
Hi-speed printer for online messages, can afford marginal guesses, use no typer.
Overprint macros, Operator notebook, man-machine relations
Self-adjusting system, fence and read-only memory.
Job log and auto-billing, SOS, Corbie checkout system. Statistics and servo-adjust.

GENERAL - BUILDING THE PROCESSOR (We must be smart too)

Object must run on same machineas processor, viz emergencies, breakdown, inordinate cost
Bootstrap, 7070 with AC III, How to bootstrap with simple origin, sortmerge, insert, etc. also table ops, programmer macros, additional functions.
Helt's comparison study, standardization
* macro in AC and 7070, guarantee retention of work checked out (illustrate) flagging, this shows one type of compiler to build compilers, special tools never used again.
Modular flow, fast and many passes, resequencing on conditionals
Check impossble memory ot unavailable equipment. Copy in line, IAL, syntactics, recursion
Expand on exception basis, pay price if wanted badly enough (illus by table of correspondence 704 (6), 705(5 or 10).
Self-repair and correction, adjust to available machine, back off to new tape updated.
Diagnostics and backtalk Reserialize, reodder and collect like elements, imp,decl,ordrs

APPLICATIONS

How individual programmer can utilize the best
Clocks and countdown time on sorts, etc.
Max allotted time for a process
Good report labeling and form (negative fathoms)
Self-identifying data, is it what is expected.
Identify units by name, notposition

HOW SMART ARE YOUR PROGRAMS?

Not only should the computer be continually busy thru the medium of a supervisory program, but it should also be engaged as much as possible in useful work. Various means are outlined for achieving this goal on present machines, plus what will be necessary on future machines to expand these techniques. (Have outline).

GUIDE - CHICAGO
6 NOV 1958

MR. BOB BEMER:

I am simply off on a sales kick. My topic for this year is, as you can see, "Intelligence in Programming Systems," and I think there is a good field here. The basis for this decision, on my part at least, is the supervisory programs some companies have written for the 704.

Take the 704 as a sample machine and say that it has an output of 1, as an arbitrary denominator. The people at General Motors Research Center, Convair, Lockheed and such have found that by controlling the complete operation of the machine through supervisory programs they are able to achieve essentially factors of about 2 in output. All of this from essentially the same machine. What you do is keep away the dial twisters, the people who think that "Just one more patch here and I can make this run," and others of that sort. You make the library accessible to everyone so no one wonders where the tapes are. In general, you get a great deal of profit out of **xxx** operating a machine which controls itself.

Now my claim for this year, and succeeding years, is that we can bump this up to a factor of 3 by making sure that not only is the machine run continually, but that all work done on the machine is useful work. That is what I mean by "intelligence in the program"; the machine itself finds out when it is doing something wrong, or it finds it is doing something that is not useful to you or that things are not just as they should be. I have a good number of examples of this. Part of this talk will be based upon a talk I gave in Denver in July and although I do not have copies here, anyone who would like one can certainly write me and I will be glad to give him one. It looks like this (Displaying). I have just one here and the man who wants it most may have it.

also
There is another paper in this area which I do not have copies of, but can tell you where to get. This is a summary of a meeting held by the Los Angeles chapter of the ACM, on the 21st of August 1958, called "Let the Computer Decide." This is a series of abstracts of the various talks at that meeting and is right in this same area. Some of the talks are "Computers and Decision-Making in Industrial Situations," "Quality Control," and "Chemical Equilibrium Program," and so forth. This is a very interesting set of abstracts and I think scientific people in particular will profit from them as there is probably more opportunity to use this type of intelligence in scientific areas than in data processing.

Now, I would like to give you some examples of what I mean by incorporating intelligence in data processing systems. The first one I thought of came from a sorting application. Some of the large sorts we have had, particularly on the 705, require a lot of tape changing, which in turn requires knowledge of when each phase of the sort is going to be through. It seems to me that a very simple way of doing this, since once can compute the time for the entire sort after the first pass (or phase) is to find out how long it is to run and to cause a count-down on the typewriter, essentially--"five minutes to go, four minutes to go, better change tape #__," etc. This is valuable because now you can cut the over-lap of manual changing by being ready to put the next tape on, make a decision and change tapes, all from this simple device, without which there would be five minutes of rushing around the machine that is absolutely wasted. This is a particular loss that must be eliminated if we are going to get efficient utilization out of our computers.

Now I would like to go through a selected list of devices that can be put into your programs right now, to make a reasonable profit. I am going to break these down.

The first item might be found in the processor. "Does it take action on an exception basis so that the programs which use the minimum facilities are processed the fastest?" This is on a priority basis and we can define this very easily in a basic processor. Suppose we wish to allow 18-character names and search for our symbols on the 709. It would take three words to contain each symbol in our referral table and processing time would be large. We would have large tables and would probably have to put them out on tape because they would overflow the memory in many cases.

However, if we took action on an exception basis, such that in the first pass we found out that the programmer did not use any names of more than 6 characters, we would automatically cut out the section which handled 18-character names and went to all the extra work, and process through the fast 6-character section. This is like the super market where you have a fast check-out for five items or less, and one makes a profit here. If you are not going to use the maximum of your facilities, operate in a cheaper way.

Another thing that concerns us greatly in making efficient programs is that one should never compile a full memory load for overlays. You always have to leave yourself room to get out into another routine, particularly supervisors. I have seen too many bad examples of people who say; "This is the way the program will be finally, so I will just run up to the last storage position because I can save a little something by using those last 12 positions." The very first change that comes around, he is stuck because he has no means of drawing another section, as an interstitial gap for buffer purposes would. There is no such thing as a final program. You must leave yourself a "pseudopod" as Dick Ridgway calls it, so that if you get trapped in a corner you have the mechanism to get out, to replace and bring something else in. This is particularly useful for supervisors.

Let's consider eliminating compiling before running. This is another place where we can make a good profit out of the machine. Suppose I have a hundred different ways of doing something; these are options. There are decisions that branch out to a hundred operations. I would rather do it on an exception basis and compile dead ends, which would lead me to the pseudo pod, which leads me to the supervisor, which brings in the processor, which takes a source language statement that is supposed to handle this situation, compiles a set of instructions, adds it to the program, puts the program back in its original condition, and now operates the routine that is necessary for this particular configuration. This is essentially a basic form of learning and one that you will profit by very much. In other words, it seems that if you have a very, very few paths that are ever taken, there is no sense in keeping all possible paths in memory at the same time when you can make a profit by having the extra room used for something else.

Here is one feature that I am very much excited about, which we are doing in our current processors. That is, the processor, before commencing, interrogates the machine to see if it has enough components to run itself properly. SCAT for the 709 will do this. Autocoder III will do this as well, and there is a further extension whereby we don't only check to see we have the information to run the processor, make sure everything is turned on and in running status, but--If the source program itself says, "When I finally create an object program run, I am going to need 5 tapes and a printer," the processor can copy this interrogation program from itself into the object program such that this must be the first segment executed in the object program.

The machine should test itself to see whether you have enough components on line in the proper status to run a program. Otherwise there is no sense of going through 20 minutes of compiling only to find out you need this tape but somebody else's old tape is on the drive so you have to stop the program while the tape is changed. This is another indication of what I mean by Intelligence. Make sure you are ready to go so you don't stop wastefully in the middle.

A case in point would be to interrogate the printer to prove that you don't have any particular type of tape control on that will be damaging. You may recall that in the PRINT system only the 12 and 1 column punches are allowed and everything is hung up if you have anything else. It was deliberately planned that way because when I get into operating a STRETCH machine, with a printer on line to give backtalk, I don't want to compute 20 minutes at STRETCH prices and find out I have left So-and-So's carriage-tapes on so it is all garbage. I just can't afford it anymore.

We have to make sure when we are ready that it will run this way. The time to check this is before you start and if you find it is not so, either take action there or get somebody else's program on the machine.

Another function of the processor is to control those sections which are designed to make the object program efficient so that they are switched out unless you are ready to run the final version. FORTRAN for the 704 is a case in point here. Optimizing index registers takes a great proportion of the time, in many cases as much as eighty percent of the process time. If one reprocesses about five or six times, it will be very wasteful. You want to turn the optimization off until you think the program is actually going to run and you need the efficiency. It may turn out that the object program may not run long enough to warrant the extra processing time. That is another decision one has to make.

Again on the processor, it should accept statistical information about frequency, to optimize the program flow. This is something some existing processors (FORTRAN) can do. In others, if one knows some things are going to occur most frequently, he can write these manually so they will run in the shortest ~~skeet~~ chain. Later on we will be building processors that will take care of this for you automatically. But be sure the program that gets the heaviest load gets the choice of components and can run the fastest.

Here is something that goes into a processor that is very useful. That is, guarding against impossible memory addresses. If you create memory addresses which are greater than the size of the machine you have or if you call for tape units which don't exist in the configuration the way you planned it, the processor should catch this before you run the problem.

Does the programmer have the option of having numbers with power underflow replaced by true zero? This is a highly argumentative point. Sometimes we come out with physical results that are so small they say, "Call these zero anyway," and other times it is vital ~~sake~~ to know when it happens. If the first man on the machine is willing to call it zero and the second man is not, you want to be sure something in the program controls which ~~xxxx~~ way you are going to do this. There should be something in my program as I drop a card or tape into the machine that ~~xxxxxxxx~~ how I am going to make this decision. Make sure to confine these various decisions in your own program.

Something very useful to keep track of your operations is when you have the supervisor produce a permanent job log and record on line. This includes the program and its name, the programmer who has written it, the date of the ~~xxxxxxx~~ run, start time, lapse time, stop time, and classification of the time (whether the run was used for debugging, production, ~~testing~~, scheduled or unscheduled maintenance), and the reprocessing number. ~~Xxxxx~~ By that I mean the number of times this same program has gone through. It may be the fourth or fifth time you have assembled this particular program; reassembly is the normal mode. The fact that you have to assemble for the first time is a special case in that there are no corrections to be made.

Another point in connection with this is that it is very useful to have a combined checkout. I refer ~~xxxxxxxx~~ you in particular to the Corbie system of the National Bureau of Standards. The specifications for this, I believe, are available through SHARE. Corbie by the way, means nothing

except they apparently copied it after a brand of liquor. It is a system where all the programs in a particular installation are kept in source form together on a single tape, numbered together with major revision sub-numbers. When a programmer wants to come in and make a correction to his program, a master program first matches the input cards he has for correction against the tape. It will search down that tape until it finds the source program, pull it out, make the changes and rewrite the source program tape on another unit. Each time a man comes in with corrections, it must copy the tape over again. That brings him up to date and in that fashion they find they misley a lot fewer cards and the tapes are not fouled up. Information can be obtained from the National Bureau of Standards or SHARE.

I have already mentioned the fact that we can instruct the processor to compile sections of program as needed on demand. I think that is a very good means of operation. We also need restart points for automatic restarts. This is something you can program in right now. If you think you are going to get into difficulty in the programming system you are using, every once in a while spill out on tape the program as it stands, including the instruction registers and that sort of information, so that in case you really get stuck on a blow-up in a long program, you don't have to go back to the beginning again. This makes a great deal of difference when you are working on a machine in marginal condition. If you find it takes three times through each problem to actually complete something, you have wasted too much machine time. If you can go back to a reasonable restart position, you will salvage all this time and get a lot more work done.

Here is a scheme that I think is very good. Can you specify, in the program, the maximum running time? Take an instance where a process should converge or produce an answer by two minutes; the program gets in a loop you don't understand and you sit disconsolate for twelve, fifteen, or twenty minutes, wondering why you don't get an answer out of it. The thing to do is say, "I want to spend two minutes on this. If it doesn't give me an answer in two minutes, go to a new program." Give a time limit on it. It saves a lot of bad programming. Of course, when you exceed the time limit, there should be some diagnostic output so you don't try it again and run into the same impasse.

Something very useful in running an installation is the facility to run continually on testing and intermingled source programs, object programs and data. Of course the supervisor should handle all of this, but you should have the option, if the program runs perfectly well, of putting on some test data or actual data and running the program then or, if not, getting off, making a diagnostic and having the next man's problem run immediately.

Something that has not been used too long, but is very useful, is the operator instruction and operator notebook concept. What this really means is that there are certain instructions which one should associate with a program. If I pick up a program two months from now and have forgotten what it was supposed to be, what the tape said, what my console switches are, etc.--the only place the identification of this should be carried is with the program itself. I don't even want it on a piece of paper associated with the program, but right in the program so when I start running the first thing that prints out is, "This program does so-and-so; you should have so many tapes; this program will check to see if they are on line, so you had better turn them on; in twenty minutes you will have to make a decision, and you will be given another warning two minutes ahead of time."

It sounds fantastic, but when you think of it, it is only trivial to program something like this. Initial printout should indicate the running control settings, what initial settings should be made manually or by control cards, what the tape assignments should be, and should even give you expected occurrences, that is, what might happen at various points. This gives you a feeling for what is going on in the machine. It is not just running.

This is akin to the audible signal on many computers. The first I saw was on SWAC at U.C.L.A. The programmer sat there knowing the topology of his program and it would play a little tune for him; as soon as the tune changed he knew he had completed a certain section of the program. If something went wrong he was able to cut it off immediately. The same thing can be done by having verbal flags in the middle of your program to tell you where you are and what you ought to be doing about it.

Another thing that operator instructions should do for you is select from a multiplicity of starting conditions. If there are different ways one can start the program, don't get hung up because you haven't done some particular thing. Let the program try this way and if it is not able to start that way let it try another one. There are many times when you will have various possibilities of starting, and the operator instructions should note which one was selected. You shouldn't have to make the decision on something like that; let the machine make it for you.

The area of diagnostics is where we do a lot of work. You want to have a tremendous amount of backtalk coming from both the processor and the running program. One of the biggest difficulties we have is scaling. The fastest way I know to stop a program in floating point operation when it is not running right is to say, for example, that when altitude > 16,000' or < 2000 feet negative it is not reasonable, because these calculations are for automobiles and not airplanes. I know certain things about the programs; I program in a trap and test the variable to see that it falls between the high and low feasible limits. Test perhaps every hundred times or something like that, so it is periodic; if anything goes wrong and it does not meet the limits, simply turn it off and say, "I am sorry, something went wrong here." I can just about guarantee in many cases in scientific computing, particularly during a check-out period, you can cut your program diagnostics in half by having this type of operation. It is a very simple test. I presume future processors will have pseudo-statements like, "Check every thousandth time to test this variable." If out of range, they will tell you what happened, insert the value in a message and do something about it.

This type of thing is what I am looking for on all of our programs in the future, tests to make sure that whatever you are running on the machine is useful work. I am sure you will agree that there is a lot of non-useful work that has been put on computers. Of course this doesn't cover the case where the user has been running a thirty-second order matrix for airplane vibration problems, and as soon as he gets a larger machine he immediately feels he needs a 60th order matrix or even 128th. This you can't control in the computer; it must be up to the good judgment of the user. But there are many cases when the operation of a computer produces invalid results or things are not going to be useful to you, and a lot of time is wasted in obtaining these. I would like to see every user feel strongly about these matters and incorporate this type of checks in his systems.

CHAIRMAN THOMPSON: Are there any questions? I have one, Bob. You mentioned time limit. You might want to try something for two minutes. Without a clock mechanism in the computer, how can you do this?

MR. BEMER: It is very difficult. However, General Motors and other people in the 704 area have built clocks for their own equipment (and some have done it for as low as twenty dollars) to give them a feedback signal to be able to do this. They demand such a clock for the 709.

There will certainly be a clock on the STRETCH computer and the 709 is set up for external trap so it is not difficult to put a clock on it and get the signal back. I don't know what the characteristics are on the 705, but you might look into it because there are so many things that can be done with a clock.

Take for instance a part of the standard SOS system for the 709. They list the programmer, a job log, a continuous log of the day of the start-stop time of each job, what is being done, the name, who is responsible for putting it on the machine, etc.; they want to keep records automatically and I don't blame them.

MR. KUSS: Yesterday afternoon a man from Commonwealth Edison gave a talk and he announced that he never bothers when he gets an 902 on the tape, he immediately stops. He doesn't have the machine read it back over three times. He said his philosophy was it is a good machine and if the tape is bad you might as well get rid of the tape. What is your philosophy? I think a lot of time people do an awful lot of work on the machines going through these routines when they are really not accomplishing much.

MR. BEMER: Well, this business changes. With the 729 units you have automatic checking, and on the 709 at least there are programs devised that will skip bad spots on the tape. If you are not able to write on a certain area of tape, you put marks to block it off and records will bypass it. This changes the philosophy. As you write you just go into a programmed routine which forever afterwards skips you over that spot and you don't have that problem.

But, if you are still on machines that don't do this, it depends to a good extent on how much work you have to go through to check something three times and how many series of three-time checks you run into. If it reads correctly only on the second time for maybe thirty times in a row, I would say something is seriously wrong and I wouldn't allow it to continue on automatic operation. These things have to be weighted and of course, the weighting itself has to be put in the program. Instead of counting the number of times we have to reread the tape, let's count the number of times we had to count to reread the tape.

MR. KUSS: You might be interested we just learned a couple of weeks ago if you stick an ordinary AC-DC radio on top of the frame and turn it on, you can get the oscillator and you can hear your program go through loops. It comes out very noisy, but some programs have a nice tune they play. I haven't done too much experimenting with this.

MR. TRUE: I would like to ask you a question. It is a little out of the range of what you were just talking about, but you did mention something about it. I noticed in Article 3, when it came out, page 6 or 7, when they got into the total point C, they had no test leading zero. I wondered if it were a misprint or the same thing you are talking about here? That was the only thing I found at all in the manual. At least, I tried to relate it to the present.

MR. BEMER: I didn't see that.

MR. TRUE: I will show it to you.

MR. HERMAN W. NELSON: Did I understand you correctly in stating some of the exceptions,

A. O. SMITH Corporation

MILWAUKEE 1, WISCONSIN

June 30, 1958

Mr. R. W. Bemer
Programming Systems
IBM Corporation
425 Park Avenue
New York, New York

Dear Bob:

I am writing on behalf of the Mid-Continent Computer Club of Chicago to invite you to speak before us. The Club, which has been meeting for over two years, is a group of mathematicians, scientists, and engineers of the Greater Chicago area who are interested in understanding and extending the technical uses of digital computers.

The Club ordinarily meets at a restaurant in the Chicago loop beginning with a cocktail hour at about 5:30, allowing about an hour after dinner for the speaker's prepared remarks. Then there is generally a considerable amount of discussion. The meetings ordinarily break up between 9:30 and 10:00. Our usual attendance is 70 to 80 people. Previous speakers have included Grace Hopper, Zdenek Kopal, A. H. Taub, and Nicholas Metropolis.

We meet every two months on the first Friday of the month. You might find it convenient to speak at our December 5th meeting since GUIDE meets in Chicago on December 3-5. We thought you might speak on a topic related to your work in programming systems.

Since the Club is underwritten completely by the members, we are unable to offer to reimburse you for your travel expenses. We hope, however, that you will be able to arrange to come. If you desire further information, you might contact Mr. Charles Swift who is the Applied Science Field Representative for this district. He is an active member of the Club.

Very truly yours,

Bob Brittenham

W. R. Brittenham, Vice-President
Mid-Continent Computer Club

WRB:ds

THE UNIVERSITY OF CHICAGO
CHICAGO 37 • ILLINOIS
THE SCHOOL OF BUSINESS

October 28, 1958

Mr. R. W. Bemer
IBM Corporation, Data Processing Division
112 East Post Road
White Plains, New York

Dear Mr. Bemer:

We are looking forward to your talk scheduled for the December 5 meeting of the Mid-Continent Computer Club. The gathering will be at Stouffer's Restaurant, 26 West Madison Street, at about 5:30 for cocktails followed by dinner.

Would you please, at your earliest convenience, send me a title for your talk and one or two sentences of description, for our meeting announcement.

Speaking on my own behalf, I saw the writeup of your Denver talk ("A Checklist of Intelligence for Programming Systems"), and would like very much to have a copy if such is available. I am at present engaged in the design of the MANIAC III at the University of Chicago, and people are now beginning to think about programming systems for it. It goes without saying that I would be glad to receive any other relevant material you may have for dissemination.

We appreciate your being able to appear before us; let me know if there is any way we can be of assistance.

Sincerely yours,

Robert L. Ashenhurst

Robert L. Ashenhurst
President, Mid-Continent
Computer Club

RLA:ecc

"How smart are your programs?"

D2 ALGEBRAIC TRANSLATION 114 DAVID KINLEY HALL

Chairman: Grace M. Hopper, Remington Rand Univac, Philadelphia, Pa.

10:20 (28) "The Construction of Algebraic Compilers,"* Alan J. Perlis, Carnegie Institute of Technology, Pittsburgh, Pa.

11:00 (29) "Translation Between Algebraic Coding Languages," Robert M. Graham, University of Michigan, Ann Arbor, Mich.

11:20 (30) "A Command Language for Handling Strings of Symbols," A. J. Perlis and J. W. Smith, Carnegie Institute of Technology, Pittsburgh, Pa.

11:40 (31) "Computer Language Compatibility Through Multi-Level Processors," R. W. Bemer and D. A. Hemmes, IBM, New York, N.Y.

13th

NATIONAL
MEETING
OF THE
ASSOCIATION
FOR COMPUTING
MACHINERY
AT THE
UNIVERSITY
OF ILLINOIS
URBANA
ILLINOIS

WEDNESDAY
THROUGH
FRIDAY
JUNE 11-13
1958

PROGRAM

SUNDAY — 3:00 - 9:00—Registration

9:00 Informal Get-Together
Extension Conference Center

MONDAY — THURSDAY — 8:30 - 10:00

Principles of Programming
Members of the Penn State Faculty

Ideas and techniques connected with the preparation of problems for solution on an automatic computer will be considered. Instruction modification, iterative loops, the use of subroutines, and index registers will be discussed.

A coding system designed both to be simple to code and to illustrate typical features of other commonly-used coding systems will be used for purposes of programming. This code can be simulated on the Penn State University computer, and problems programmed in this code can therefore be run directly on the computer.

MONDAY - WEDNESDAY — 10:15 - 11:45

Components and Characteristics of Computers
Kenneth Powell, I. B. M. Corporation

The nature and function of the various parts of a computer will be developed from simple ideas. Different types of components will be discussed and related to the requirements of their use.

MONDAY AFTERNOON

1:15 Principles of Programming

3:00 Programming Workshop
The seminar will be split into small groups and members of the Penn State faculty will be available to advise on programming the problems which will subsequently be run on the computer during the computer workshop periods.

7:30 Computer Demonstration
The Penn State computer will be demonstrated and the actual solution of a simple program on the computer will be carried out in slow motion.

9:00 Coffee Hour

TUESDAY AFTERNOON

1:15 Programming Workshop

3:00 Picnic Supper
An excursion bus to one of the surrounding beauty spots.

7:30 Computer Workshop
During this time the computer will be available to students for the solution of problems.

WEDNESDAY AFTERNOON

1:15 Numerical Techniques for Computers
Mary Lister, Assistant Professor of Mathematics, Penn State
A survey of numerical techniques which arise in the solution of problems by computers.

3:00 Programming Workshop

7:30 Computer Workshop

THURSDAY

10:15 Internal Operation of the Penn State Computer
Robert W. House, Instructor in Electrical Engineering, Penn State.
The processing of instructions and data inside the Penn State computer will be described, and the way in which the operation of the various parts of the computer are coordinated will be discussed.

1:00 Fortran
John Backus, I. B. M. Corporation
A description of an automatic coding system in which coding is performed by writing out the algebraic expressions which it is required to evaluate.

2:30 Automatic Coding and Programming
Grace M. Hopper, Remington Rand Univac Division
A description of an automatic coding system in which the program can be specified by writing out English phrases. This method of coding reduces the time required for training coders and is particularly suited to business applications. Future trends in automatic coding will be discussed.

4:00 Engineering Applications of Computers
R. W. Ferguson, Westinghouse Corp.
The computer installation at Westinghouse will be described, and the use of computers in systems engineering and product design will be surveyed.

6:30 Banquet
Speaker: John W. Mauchly, Remington Rand Univac Division
Computers, past, present and future

FRIDAY

8:30 Programming Systems for Electronic Data Processing
R. W. Bemer, I. B. M. Corporation
A description of some of the techniques of commercial electronic data processing.

10:15 Impact of Computers on Engineering Problems
D. T. Bell, Bell Telephone Laboratories
Problems in the field of Transmission Networks Design are used to illustrate the advantages gained by the use of high-speed effects of computers on design procedures.

1:00 Learning Programs for Computers
Alex Bernstein, I. B. M. Corporation
A survey of learning experiments on computers and a more detailed description of one or two specific learning techniques.

2:30 The Playing of Games by Computer
Arthur L. Samuel, I. B. M. Corporation
A description of a checker playing program and a discussion of those aspects of the program which can be said to involve learning on the part of the computer.

4:00 Economic Application of Computers
Julius Shiskin, Bureau of the Census

7:30 Round Table Discussion
General discussion of points of interest arising out of the seminar.

SATURDAY

9:00 Optional Computer Workshop

Summer Engineering Seminars 1958

* Introduction to Computer Programming
June 15 - 21

THE PENNSYLVANIA STATE UNIVERSITY

Application Form

Introduction to Computer Programming

June 15 - 20, 1958

Kindly reserve places for the following persons:

Name and Title or Position	Check here if room is desired in residence hall: Single	Double
1.	( )	( )
2.	( )	( )
3.	( )	( )

Company and Mailing Address .....

CHECK(S) in the amount of $..... attached (fee of $60 per person) payable to The Pennsylvania State University.
Mail to: Engineering Seminars, Extension Conference Center, The Pennsylvania State University, University Park, Pennsylvania

PURPOSE

During the last few years the digital computer has become an increasingly important factor in our lives. This course is intended for those who want to familiarize themselves with the computer field. It will be of particular benefit to persons from industry and commerce who would like to evaluate the potentialities of the computer approach. No previous knowledge of computers is assumed.

This course will present the fundamental principles of programming for a stored-program digital computer and a selection of both scientific and business applications, in order that a person who has no previous knowledge of the uses of a digital computer may be able to judge whether his industry has need of electronic equipment. If a computer has already been acquired, this course will provide a basic training for potential users of the machine.

Students will be encouraged to program a simple problem in their own special field of interest on the Penn State University computer. Speakers from leading computer firms and from industry will describe the characteristics of their computers, and their applications to specific problems. Automatic programming techniques and "learning" by computers will also be discussed.

Throughout the Seminar there will be opportunities for personal contacts with persons who have a broad experience in the programming field.

MEETINGS

The Seminar will be conducted mainly on a lecture-discussion basis. The first three days will be taken up primarily with teaching the fundamental principles of programming, and the last two days with talks by outside speakers.

LIBRARY FACILITIES AND EQUIPMENT

The facilities of the computer laboratory library will be available during the Seminar. There will also be a special library of recent publications in the computer field. The University's computer and data processing equipment will be available for use during the course.

Sales representatives of leading computer firms will be available for informal discussions.

SEMINAR ON INTRODUCTION TO COMPUTER PROGRAMMING

June 15 - 20, 1958

The Pennsylvania State University

College of Engineering and Architecture

General Extension

Seminar Chairman, Peter Wegner

Research Assistant in Engineering Research

with cooperation from

International Business Machines Corporation

Sperry Rand Corporation

REGISTRATION

The registration fee for the Introduction to Computer Programming Seminar is sixty dollars ($60). This includes all costs of instruction and materials, and a certificate of attendance. Charges for housing in campus residence halls are additional and are payable upon arrival.

The University reserves the right to limit total enrollment to the maximum which can be accommodated in available classrooms or residence halls.

Application forms should be accompanied by a check, money order, or proper authorization (military personnel) for the registration fee, payable to The Pennsylvania State University. Refunds will be made for cancellations received prior to the start of the Seminar.

HOW TO REACH THE UNIVERSITY

Penn State is in the Borough of State College, Pennsylvania, near the geographical center of the State. It is on highway routes 45 and 322, and may be reached in less than six hours by car from any part of the State. The community is on the east-west route of the Greyhound and Edwards Lakes-to-Sea bus lines.

Those traveling by train (Pennsylvania Railroad) should get off at Lewistown (from the east), and at Altoona (from the west) where bus connections to State College are available.

Allegheny Airlines serves central Pennsylvania, stopping at Black Moshannon Airport where limousine or cab service to the campus is available.

ACCOMMODATIONS

Persons attending the Seminar are invited to reserve rooms in our well-equipped University residence halls. If you are interested in accommodations of this type, will you please make your reservation at least two weeks in advance?

Double rooms with twin beds and a very limited number of single rooms are available. Because of the limited number of single rooms, it will be necessary to assign two people to each double room. If you wish to room with another person, note his name on the room reservation form.

The residence hall rooms do not have private baths; however, well-appointed bathrooms are located conveniently near the rooms.

The room rent is $3.00 per night per person in double room and $3.50 in a single room.

If you have checked into the residence halls and find that you must leave before the end of the Seminar, you will be entitled to receive a refund on full-day increments.

Check-in time for the residence halls is between the hours of 2 and 9 p. m. on the date of registration. If you are not able to arrive during these hours, please notify us so that we can make convenient arrangements for you.

Meals may be secured at the Hetzel Union Building on the campus, or at the numerous restaurants in the community.

Other accommodations may be secured at the Autoport Motel, College Court Motel, Horlacher's Motel, Nittany Manor Motel, Nittany Lion Inn (University-operated), Ranch Court Motel, Penn State Motel, State College Hotel, Taylor Motel, or Travelers Lodge by writing directly. The address for all is State College, Pennsylvania.

UNIVERSITY OF DENVER Denver Research Institute
COLORADO SEMINARY
University Park, Denver 10, Colorado

31 July 1958

Mr. R. W. Bemer
International Business Machines Corporation
590 Madison Avenue
New York, New York

Dear Mr. Bemer:

Thank you for the very fine paper you presented at our symposium and for your contribution to the panel discussions. Your participation as speaker and panel member added a great deal to the success of the occasion.

So far we have received a number of favorable, as well as some disparaging remarks on the conference as a whole. To help us with our plans for next year, we would appreciate greatly your suggestions or criticisms.

I sincerely hope you will join us again next year for the sixth in this series of symposia.

Yours very truly,

C. A. Hedberg, Head
Electronics Division

CAH:ch

benson-lehner *corporation*

applied cybernetics

28 July 1958

Mr. R. W. Bemer, Manager
Programming Systems
IBM Corporation
590 Madison Avenue
New York 22, New York

Dear Mr. Bemer:

I would like to tell you how much I appreciate your participation in the panel on Systems Organization at the Denver Symposium. From what I have heard, it appears that the session was considered to be quite a success.

Thank you for your contribution to the panel.

Sincerely yours,

BENSON-LEHNER CORPORATION

Bernard S. Benson

Bernard S. Benson
President

BSB/ll

11930 West Olympic Boulevard, Los Angeles 64, California
BRadshaw 2-3484 GRanite 9-3723 Cable Address: BENSON Los Angeles

THOMAS 3035
THU MORN 26
PDB4C — UNIVARAC?

UNIVERSITY OF MICHIGAN
ANN ARBOR
DEPARTMENT OF MATHEMATICS

June 4, 1958

Mr. John Backus
Mr. Robert Bemer
IBM Corporation
World Headquarters
Madison Avenue
New York, New York

Dear Mssrs. Backus and Bemer:

This is to invite you to attend our conference on Artificial Intelligence and Digital Computer Programming, June 16-27.

In any case the University will waive the tuition charge. If you should stay just a few days, I would not expect any activity from you unless you desire to participate. If you were to stay the entire period I would ask you to give a report on your latest activities, either as a lecture, or as part of a panel discussion.

Please let me know if you can attend, and if so, when we may expect you so that we can arrange housing. We are planning a Saturday discussion similar to last year's which might interest you. In addition, four Soviet scientists are scheduled to be here, as well as J.H. Wilkinson of Great Britain and Paul Lorenzen of Western Germany.

Sincerely yours,

John W. Carr III

John W. Carr III
Assoc. Prof. of Math.

JWC/md

benson-lehner *corporation*

applied cybernetics

27 May 1958

Mr. R. W. Bemer, Manager
Programming Systems
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

Dear Mr. Bemer:

I have been asked by Mr. W. H. Eichelberger to act as chairman of the panel on Systems Organization at the Fifth Annual Symposium on Computers and Data Processing at the Denver Research Institute on July 24 and 25. The list of panel members made it so intriguing that it was impossible to refuse, and I get the feeling that we are going to have an extremely interesting time. It might even be interesting for the observers!

In this direction, I feel that we achieve and contribute most as we focus on areas where we have divergent views, on the basis that "If in a discussion two of us agree, one of us is redundant!" I feel that if we can have a menu of about five really difficult and controversial questions, this is just about all the preparation we need to do.

I would like you to contribute your thinking to this list, if you would be so kind - questions such as:

1. Which is usually the area of greatest incompetence in systems organization work?

2. Which end should be considered the beginning of a system - where the information goes in or where it comes out?

Could you let me have your thoughts, on a very informal basis, and I will send you back the menu so that you can be chewing on it ahead of time. I am looking forward to this panel with a great deal of interest. I will be very happy to receive any suggestions which you might have to enhance its value, including changing the chairman!

Sincerely yours,

BENSON-LEHNER CORPORATION

Bernard S. Benson (ll)

Bernard S. Benson
President

BSB/ll

11930 West Olympic Boulevard, Los Angeles 64, California
BRadshaw 2-3484 GRanite 9-3723 Cable Address: BENSON Los Angeles

UNIVERSITY OF DENVER Denver Research Institute
COLORADO SEMINARY
University Park, Denver 10, Colorado

16 May 1958

R. W. Bemer, Manager
Programming Systems
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

Dear Mr. Bemer:

Since writing you on 17 April, we have completed preparation of the program for our Computer Symposium. We believe the worth of the technical sessions would be enhanced if some of the speakers were to participate in the panel discussions in addition to presenting their papers. Therefore, we have taken the liberty of including your name in the list of members of the Systems Organization Panel. This panel discussion will take the place of part or all of the question period following the papers.

The chairman of the panel session will be Mr. Bernard S. Benson, President of the Benson-Lehner Corporation. He intends to contact the panel members prior to the Symposium for some preliminary organization of the discussion. Other members of the panel in addition to yourself will be the following:

Dr. Ernest A. Keller, Director, Research Division, Panellit, Inc., 7401 North Hamlin Ave., Skokie, Illinois
Dr. Paul Brock, Statistical Laboratory, Purdue University (On leave of absence. Present address: P.O. Box 453, Monterey, Calif.
Jerome Rothstein, Edgerton, Germeshausen and Grier, Inc., 160 Brookline Avenue, Boston 15, Mass.
John F. Bishop, Gen. Mgr., Beckman Instruments, Inc., 2500 Fullerton Road, Fullerton, California
F. M. Verzuh, Asst. Dir., Computation Center, MIT, Cambridge, Mass.

We trust these arrangements meet with your approval. We are looking forward to seeing you at the Symposium.

Sincerely,

W. H. Eichelberger

W. H. Eichelberger
Chairman, Papers Committee

WHE:E

23 JUL
→ UAL 735 4 PM IDL — D → 700 DEN

25 JUL
UAL 748 2PM DEN — D → 1010 IDL
UAL 749 945 AM FDL → 1245 DEN (ALTERN

RIDER'S ?

6/4/58

XTRAN ANNOUNCEMENT - SHARE Meeting - 26 Feb 1958

XTRAN is a tentative name for a tentative source language which is to be a superstructure on the existing FORTRAN language. Certain of the preliminary specifications are outlined on the sheets you now have, together with some coding examples to demonstrate certain salient features. Note that these specifications are incomplete, particularly with respect to input-output and logical statements. This does not imply that we do not have improvements developed, but merely that we could not decide on a proper form for this presentation, rushed as we are.

XTRAN follows a method demonstrated to be feasible by the FOR TRANSIT system, which is a means of running source programs in the FORTRAN language on the 650. This is done by means of a source language - to - source language processor from FORTRAN to the IT language, which is then used to produce symbolic and eventually machine language coding. Thus XTRAN programs produce FORTRAN programs which are then fed into the FORTRAN processor for the 709. Again, the standard SHARE machine is the configuration used thruout.

The basic mechanism of XTRAN is the Pre-Processor program, which is being currently flowcharted for the 709. This pre-processor is itself a multi-pass intelligence-gathering and transforming unit quite similar in principle to the multiple passes of the PACT system. I believe that most of our future programming systems will be constructed on this modular principle, limiting the variety of functional work performed on each pass so there is little conflict, and certainly more flexibility for improvement and change. Thus the information gathered in passes 3 and 5 might show that there were no source entries requiring the services of pass 8, which would be eliminated accordingly. For processing XTRAN on the 705 models I, II and III, no pre-processor as such is required because the combined GUIDE-IBM working group is starting from this language and translating directly to Autocoder III.

During the pre-processing, many of the XTRAN statements will produce multiple statements of the FORTRAN variety, so that the program is likely to be much expanded when entering the FORTRAN processor. We are coordinating XTRAN work with that of the FORTRAN 709 processor so that the analysis and information-gathering done in the XTRAN pre-processor is switched off in the FORTRAN processor and not duplicated any more than necessary. If there had been sufficient time to make an integrated system before delivery of the first 709s we would have done so, but the present mode of fabrication has a greater safety factor for completion on time. In actuality we expect the overlap to be negligible.

The basic intentions for XTRAN are:

1. To minimize the amount of actual writing and coding, resulting in fewer entries.

2. To minimize possible coding errors by allowing more freedom in rules and automatically inserting new and corrective statements. Thus:

   a. Algebraic statements may have mixed expressions containing fixed or

floating point variables or constants. Multiple sinks may be specified, as "RUNNINGSUM = INITIAL = some expression". This same statement may be used to create multiway programmed switches.

b. Declarative statements (as opposed to Imperative statements which have to do with program flow) may be written anywhere in the program. The pre-processor automatically moves these to the beginning in the output listing and groups them appropriately.

c. Conditional transfer statements need show referents for only those statements which are not the next following. Thus IF (a) ,, THERE would mean go to the next statement if the value of $a \leq 0$. This also occurs for defining sets of instructions from the first statement to the last, in that the first need not be mentioned if immediately following. This minimizes the need to think up different names for statements.

d. Statement referents and variables may be assigned virtually ANY 10 character alphanumeric name (numbers are still a proper subset of referents). The pre-processor converts all of these to numeric equivalents for FORTRAN and assigns and creates new names as necessary. This eliminates several opportunities for FORTRAN errors, in that there need be no special names for classes of variables such as fixed point, is less danger of running over the 6-character limitation and names may be more meaningful and differentiated with less chance for duplicates.

e. Subscripting and range control on DO statements are both general and compound, allowing essentially complete freedom to use subscripts and incrementation as any fixed point algebraic expression, with the present limitation that they not contain parentheses. Here again is elimination of error source and added convenience, particularly in the use of negative subscripts and decrementing in a DO statement. Variation for all indices may be expressed in a single DO statement and each of these may have multiple stepping, as a(b)c(d)e..... (See the matrix multiplication ex.)

f. Most fixed point variables are determined to be such from their usage in the statements. Those few which may not be are so stated in a FIXED POINT LIST statement. This also applies to Boolean variables.

g. Much writing may be eliminated by the DEFINE and EXECUTE SET statements, which effectively copy groups of coding in place.

h. Many misspellings and incorrect statement punctuation will be detected and automatically corrected and ignored. Those which cannot are at least detected before they go into the lengthier FORTRAN processing, as are logical errors in coding and ambiguous statements.

3. To be compatible with lower level existing languages in that any FORTRAN statement may exist in XTRAN, but not vice versa. This is important in that there may be cases where XTRAN programming when used to the utmost would produce a less efficient and slower-running program, although programmed much more simply in source language. In this case, as with the present operation of 704 FORTRAN, the expert programmer may achieve better efficiency thru understanding of the system and coding in the lower level terms.

4. To be compatible with many IBM computers with like capability, such as the various 705s, Tape 650, etc. Machine orientation has been removed from the language and facility is provided to code specific sections in the symbolic machine language of the several computers, meanwhile maintaining a careful watch to catch and give warning if such programs are run on other computers for which these sections are unacceptable.

5. To allow the fabrication of generalized higher-level statements by the open-ended definition of new language with the appropriate generators. For this it must have recursive properties and be susceptible to set notation, symbol substitution, logical algebra, etc.

6. To allow a more nearly "flowchart" type and way of coding, where the detailed blocks of coding are filled in later although not subjected to restrictions that they must appear in the source program in any particular flow pattern.

7. To facilitate multiple processing and testing of intermingled programs

PRINT EXAMPLES OF SYSTEM CONTROL
SYSTEM TAPE BURIED IN MACHINE,

NOTES ON FUTURE AUTOMATIC PROGRAMMING

The characteristic of future computors of immediate concern to programmers is that they will not be able to have direct and personal access to the machine. Such access is prohibited by speed and organizational characteristics. It is obvious that a central console is similarly useless; human usage of START and STOP buttons, alteration switches and the like would drastically detract from the efficient usage of the computor, by factors of up to 5 to 1 in the case of superfast arithmetic units. Consoles will exist only in a limited form on the various pieces of peripheral equipment.

Since the machine cannot be controlled by a human, it must control itself thru a supervisory program which exists in a higher level than the subject programs. This program must be capable of:

1. Programmed control of and communication with the peripheral equipment.
2. Scheduling the various problems presented to it. The future computor might be likened to a short order cook, the peripheral equipment to his order drum and the various people with problems to waiters placing their various orders on the drum. A cook does not necessarily take orders and process them in order of placement in time. He tailors the operation to the present and future loading of his facilities. The supervisory routine, upon completion of each job, must inspect all current orders and decide which to process next, depending upon the components required. It might even decide to temporarily delay a long problem to do several quick jobs in a row, to make the most ~~numbers of~~ programmers happy. These features are of course recognizable as applications of SHOP SCHEDULING and QUEUE THEORY, for which mathematical models and algorithms are being developed.
3. Interrogation of a control section which each problem should supply, which specifies limits within which certain answers are reasonable. If they are not, the supervisory routine must determine, again from these specifications, what type of information should be sent to the peripheral equipment for diagnostic purposes.
4. Being overridden. The supervisory routine must be capable of supplying information where missing. When such information is supplied by the program, it takes precedence over what the supervisor would supply.
5. Calculating approximate execution times for the various programs, with servo-correction at discrete intervals, in order that peripheral equipment and the programmer may be alerted. This might be extended to ringing the programmers telephone.
6. Transferring operation to different components when errors are detected. The type of error should be well analyzed and reported to the engineers.

Current high programming costs must be reduced by producing general synthetic languages not specifically apllicable to any machine, but capable of translation to the language of a specific machine by a program operating on that machine. In certain rare cases it may be practical to produce on one computor a program capable of being operated on another, but this permutability can be a monster if uncontrolled and should be avoided.

Programs such as FORTRAN can greatly reduce the amount of programming time necessary to prepare problems. FORTRAN, under field conditions, has reduced time by a factor of 6. COMTRAN is expected to achieve a factor of 10 or more. However, huge sums will still be wasted unless the ideal conditions are achieved; namely, that the automatic programming system must be deliverable to the user at the same time as the machine itself.

In order to achieve this desirable condition, a tremendous spurt in automatic programming effort must be made right now. Otherwise we will never be able to emerge from stop-gap measures which cost more money and energy than it would to develop proper original programming methods.

Since there will be no obvious need for anything other than the common synthetic language, it is probable that future programmers will never need to learn machine language. If the translation program incorporates a high enough level of intelligence, it will be able to devise a machine language program as efficient, or more so, than the best hand programmer. This has been proved in the first usage of FORTRAN at United Aircraft.

These synthetic languages will be standard for an entire group of computors, and may well prove to be the best comparison and evaluation method between different designs. Since only the machine manufacturers personnel will work with machine language in the fabrication of the translator, it is no longer necessary to design a machine language for coding facility. Future machine languages should be designed for , and their form dictated by, optimum speed of operation. Micro-programming is a form this may take.

It is likelythat diagnostic techniques will be altered greatly from the methods we use now, probably taking the form of complete recoding or restatement of the problem in the synthetic language, without reference to the original abortive try.

A new class of personnel, midway between an engineer and a programmer, may spring up. This machine analyst would be charged with the diagnosis of machine difficulties by test programs. Such programs will be much more complex than at present, and it will be difficult to train engineers for diagnosis, as at present.

The translator must carry analytical intelligence to detect breaking of rules or form in the synthetic language statements. This may be automatically corrective, or call for reprogramming action. A good ideal would be to have it generate a range of test data.

April 27, 1959

Dr. W. F. Bauer
Thompson Ramo Wooldridge Inc.
Ramo-Wooldridge Division
P.O. Box 90334 Airport Station
Los Angeles 45, California

Dear Walter:

The symbol manipulation business is exploding in the computer world. You have probably heard of John McCarthy's excellent conference at MIT. The effect is felt not only by these thinkers but by the actual producers of programming systems in their everyday work. I believe that concepts in this field will allow us to break through the production barrier that now exists.

We find ourselves in a situation where a movement well on its way has not been documented adequately in technical journals. I suggest a two-part initial remedy. You might ask:

1. McCarthy to write a report on the MIT meeting.

2. Someone to prepare an expository paper on symbol manipulation for the Journal.

It sounds like I have just awakened to the fact that we have wasted time in this area. This is somewhat true, but I think most people are still unaware of the impact. Let's rush to shake them from their stolidity.

Very truly yours,

rwb/ep

R. W. Bemer, Manager
Programming Systems

cc/ Dr. M. L. Jansson--The RAND Corporation
Dr. John McCarthy--MIT
Dr. A. J. Perlis--Carnegie Tech

Memorandum

DPDHQ
April 24, 1959

Mr. A. L. Harmon

Symbol Manipulation Meeting--MIT
16-18 April

Several of the very best computer men in the Country were present at this meeting. Included were:

George Mealy	Bell Laboratories
Phil Bagley	Lincoln Laboratories
Herb Gelernter	IBM Research
John Backus	IBM Research
Nat Rochester	IBM Research and MIT
John McCarthy	MIT
John Carr, III	University of North Carolina
Victor Yngre	MIT
Tony Oettinger	Harvard
A. F. R. Brown	Georgetown
Joe Smith	Technical Operations
Alan Perlis	Carnegie Tech
Doug Eastwood	Bell Laboratories
Doug Ross	Servomechanisms Laboratory, MIT
Phyllis Fox	MIT
Jim Byrne	Lincoln Laboratories
Alan Tritter	Lincoln Laboratories
Julien Green	IBM DP
Anatol Holt	University of Pennsylvania
W. Turanski	" "

During Holt's presentation on the Siggal Corps R & D, the discussion veered from UNCOL through Bootstrapping into machine-to-machine translation. I think it highly significant that not one member of this group disagreed with the proposition that it is for all practical purposes impossible to translate a program written in the language of one machine to a corresponding program in another machine language. Far from being non-committal, several members were actually vehement in stressing this point. All agreed that information lost during the translation processes is the major stumbling block.

rwb/ep

R. W. Bemer

cc/ Dr. C. R. DeCarlo
Mr. J. C. McPherson-WHQ

COMPUTATION CENTER

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE 39, MASSACHUSETTS

April 10, 1959

Mr. R. W. Bemer
IBM Corporation
Applied Programming
112 E. Post Road
White Plains, New York

Dear Bob,

Enclosed is an agenda for a conference on Symbol Manipulation. I was shocked to discover that I had left your name off the earlier mailing list.

It would be fine if you or Julian could make it.

Best regards,

John

John McCarthy

JMcC/caf

Enclosures

CONFERENCE ON SYMBOL MANIPULATION SYSTEMS

1959 April 16-18, MIT

Where: The Research Laboratory of Electronics Conference Room Room 26-217 except for Thursday and Friday afternoons when the meeting will be held in 24-502.

When: Start at 10A.M., Thursday, April 16. Tentatively, there will be morning sessions starting at 10A.M. and afternoon sessions at 1:30P.M. and evening sessions if necessary.

TENTATIVE SPEAKERS LIST AND SCHEDULE

Thursday

V. Yngve, MIT
A. Holt and W. Turanski, Remington Rand UNIVAC
G. Mealy, Bell Telephone Laboratories

CAT. AHAB & MOBIDIC

Friday

H. Gelernter, IBM
J. McCarthy, MIT
N. Rochester, IBM and MIT
J. G. Kemeny, Dartmouth College

Saturday

J. W. Carr, University of North Carolina
A. J. Perlis, Carnegie Institute of Technology
D. Ross, MIT
M. P. Barnett, MIT
A. Tritter, Lincoln Laboratory, MIT

We have abstracts as follows:

V. H. Yngve, MIT

The COMIT system is designed to enable a linguist unacquainted with mathematics or computer technology, to apply the computer to mechanical translation research with ease. We first devised a problem oriented notation system which combined usefulness, convenience and simplicity. Only after the details of the notational system had been worked out to our satisfaction did we turn to the question of programming the system on a computer. We thus have a thoroughly user-oriented system.

A. Holt, Remington Rand UNIVAC

1. Purpose

The ACT System is a programmed adjunct to a general purpose computing machine (specifically, MOBIDIC) whose purpose is to facilitate the initial encoding and subsequent application of specific code-to-code translation procedures. The phrase "code-to-code translation" is primarily meant to cover the conversion of problem oriented pseudo-codes into machine code equivalents. The ACT system provides for the "housing" of many distinct translation procedures - carrying algebraic, data-processing, simulation languages, etc., etc. into any of the FIELDATA machine codes - within the bounds of a single, controlling system.

2. Functions

The ACT system consists of a collection of programmed "components" which play a fundamental role in assisting translation. These components fulfill several kinds of functions.

2.1 Sequence control of complex translation procedures. Every translation procedure involves the action of many "translation processes". From specially encoded information about the interconnection of parts, the ACT system controls the proper sequencing of steps.

2.2 Input-Output Control - Individual translation processes call for their input lists and deliver their output lists by means of specially encoded pseudo-commands. Certain portions of the ACT system retain responsibility for making the inputs available on demand, and disposing of the generated intermediate outputs.

2.3 Fundamental Translation transformations - Analysis has shown certain data transformations to be of recurrent utility in a wide variety of code-to-code translation procedures. Such transformations, which are few in number (and are to be discussed in greater detail below) are to be prepared in coded form as part of the ACT system.

2.4 Assembly - In the manufacturing of machine code, there is a distinctive stage through which translation processes almost always pass, defined as follows. Previous translation steps have yielded an intermediate code form from the bulk of which is conformal to machine code but differs from the latter in addressing, ordering, and some other features. This intermediate code form is called "assembly code" and its conversion into machine code is called "assembly". In accomplishing assembly, the ACT system makes use of components of functional types 2.1, 2.2 and 2.3 above, as well as some special assembly components. Given an assembly coded problem for machine A, the assembly system of ACT can produce machine code A, for a wide range of A's. It does this by drawing on a library of machine instruction formats as well as on assignment routines specialized to particular computers.

3. ACT Core and ACT Library - The basic components of the ACT system - some of which control others - themselves control large collections of further components, called "libraries". There is no clear boundary line, in principle, between the libraries of components controlled by ACT and the so-called basic components which constitute the core of the system. There is, however, a practical distinction between them. The core components - or "core functions" - of the system are to be prepared prior to the specification of any particular source pseudo-code or any target machine code. The subsequently constructed components, which are determined by the language and machine specifications, are counted as belonging to the ACT library.

4. ACT Core Components

4.1 Translation Formula Interpretation - Highest level translation control (which is said to be control of a "translation procedure") resides in an interpreter which decodes a formula specifying what "translation routine" is to operate on what data and in what order. The only function of this interpreter is to recognize the hierarchical structure of the

formula and to institute these major translation steps in an allowable order.

4.2 Process Chart Interpretation - Each translation routine is represented by a process chart which shows how specific "translation processes" are interconnected. The process chart interpreter must locate and initiate each process at the proper time, as well as making the proper inputs available and storing the resultant outputs.

4.3 Call Interpretation Cycle - The function of the call interpretation cycle is to submit a string (input) to the action of a collection of stored translators. More specifically, the input consists of items containing ordered parameters and "call selectors". The items are assumed to be order-free so that they may be reshuffled for the sake of efficient call selection and execution. The call interpretation cycle delivers all the items with similar call to a suitably selected library package for transformation. The transformed result will be scanned for the possible presence of further calls.

4.4 Sorting - A translation process whose function is to re-order generated lists. The inputs will be assumed to have keyes constructed according to standard conventions.

4.5 Reduction - A translation process which "reduces" chains of definitions to a listing of initial undefined symbols against their final equivalents (eliminating all intermediate steps). (The reduction process utilizes sorting).

4.6 Symbol Transformation - A translation process which modifies "symbols" in a string by substitution, prefixing or suffixing. Normal inputs to this process include a string to be examined in which symbols are recognizeable by conventionally determined signals, a substitution table, and/or some specially encoded rule for symbol prefixing or suffixing.

5. Implementation

The ACT system is to be programmed initially for the MOBIDIC computer. The programming will most likely be done in a suitable MOBIDIC assembly language.

G. Mealy - Abstract by J. McCarthy

Dr. Mealy will describe IPL as developed by Newell, Simon and Shaw, in version for the IBM 704 which he is completing. He also is prepared to defend the following theses:

"Thesis I: In any situation in which well-formedness is defined recursively, the test for well-formedness should also be viewed as a rule for compilation.

Thesis II: Under the same conditions, a goal-subgoal structure is the natural structure for the compiler, with goals corresponding to syntactic types.

This is to say that I can visualize the first stage of compilation for IAL as being, done by an IPL-type program. The major goal corresponds to a sentence:

$(\mathcal{E})\sim$ begin Σ; Σ; m; Δ; Δ; Σ; m; Σ end

This goal says, essentially: "Is the string a sequence of statements and declarations? If so, I will compile it and say so. If not, I will tell my supergoal that this is not well-formed by my criteria." It keeps setting up the subgoals (Σ) or (Δ) (in most cases it knows which to try by knowing the next delimiter below its level). Similarly, (Σ) sets up subgoals such as (E), (I), etc."

H. Gelernter, IBM

A compiled computer language for the manipulation of symbolic expressions organized in storage as Newell-Shaw-Simon lists has been developed as a tool to make more convenient the task of programming the simulation of a geometry theorem-proving machine on the IBM 704 high-speed electronic digital computer. Statements in the language are written in usual Fortran notation, but with a large set of special list-processing functions appended to the standard Fortran library. The algebraic structure of certain statements in this language corresponds closely to the structure of an NSS list, making possible the generation and manipulation of complex list expressions with a single statement. The many programming advantages accruing from the use of Fortran, and in particular, the ease with which massive and complex programs may be revised, combined with the flexibility offered by an NSS list organization of storage make the language particularly useful where, as in the case of our theorem-proving program, intermediate data of unpredictable form, complexity, and length may be generated.

J. McCarthy, MIT

The Lisp Programming System

A programming system called LISP (for LISt Processor) has been developed. Its purpose is to facilitate programming manipulations of symbolic expressions.

The present status of the system is as follows:

1.1 The source language has been developed and is described in a number of memoranda of the Artificial Intelligence Group.

1.2 About 20 useful subroutines have been programmed in LISP and hand-translated to SAP and checked out on the IBM 704. These include routines for reading and printing list structures.

1.3 A routine for differentiating elementary functions has been written. A simple version is checked out and a more complicated version which can differentiate any function when given a formula for its gradient is almost checked out.

1.4 A universal function apply has been written in LISP, hand-translated and checked out. Given a symbolic expression for a LISP function and a list of arguments it computes the result of applying the function to the arguments. It can serve as an interpreter for the system and is being used to check out programs in the LISP language before translating them to machine language.

1.5 Work on a compiler has been started. A draft version has been written in LISP which is presently being discussed before it is translated to machine language or checked out with apply.

1.6 The LISP programming system has been shown to be based mathematically on a way of generating the general recursive functions of symbolic expressions. This appears to be a more natural approach to recursive function theory than either Turing machines or the recursive functions of integers. The proofs of the important elementary results are much simpler and involve no artificial constructions.

N. Rochester, IBM and MIT

Engineering Calculations in LISP

The application of LISP to the calculation of properties of linear passive networks is being studied by N. Rochester, S. Goldberg, C. S. Rubenstein, D. J. Edwards, and P. Markstein. A series of programs in LISP is being written. These will enable the 704 to accept a description of a network in literal terms and calculate literal expressions for various characteristics of the networks.

The equivalent two-terminal-pair network is among the characteristics to be computed. This will be done by organizing the data into a matrix, inverting it, doing a little more algebra, and then simplifying the result. For a general network of reasonable size, such a result would be too complicated for human consumption. However, if the values of the circuit parameters are all simple multiples of a few values the result is often simple while the process of reaching it may be too lengthy to be carried out by a person. It is expected that the machine can be made to produce simplified equivalent circuits to describe the actual network to a given degree of accuracy for a given special purpose. The general class of computations which are the target of this investigation are those which have a reasonably algebraic expression as an answer but which require lengthy algebraic manipulation.

J. G. Kemeny, Dartmouth College

Abstract by J. McCarthy

Prof. Kemeny will describe the Dartmouth work on a program to solve differential equations analytically and perform other symbolic manipulations.

J. W. Carr III, University of North Carolina

Extension of Present Algebraic Language Compilers to Symbol Manipulation

It has been shown that present algebraic language compilers can be used successfully for symbol manipulation, and that list structure procedures can be developed using algebraic compilers with recursive subscripting (for example, IT). The use of the standard IPL-type languages, as well as the GP-type languages, has been delayed because both of them are hard to learn. Is it possible to use the basic translational structure of the algebraic languages, combined with extensions that allow intermixed translation and interpretation, along with running, and symbol manipulation of the performing program itself, to perform many of the techniques developed, for example, with IPL and at the same time permit ease of descriptions? Some simple experiments indicate possibilities.

A. J. Perlis, Carnegie Institute of Technology

I am going to discuss a specific data processing language with emphasis on programming algorithms. The algorithms will involve analyses of boolean expressions, constructions of program sequences and best packing of data. Realization of these algorithms in an existent assembly system will be described.

M. P. Barnett and E. J. D. Carter, MIT

The Use of Fortran in String Manipulations

The Fortran coding system is being used successfully as a vehicle for developing programs for string manipulation. These are being applied to the editing and recursive generation of Fortran source language programs, to the editing of input data, to the manipulation of mathematical expressions and the encoding of programs that will recognize syntactic categories from English text.

A. Tritter, Lincoln Laboratory, MIT

The Conditions a Compiler Must Satisfy so it Can Compile itself.

IBM

(FOR INTRACOMPANY CORRESPONDENCE)

TO OFFICE AND/OR DEPARTMENT: WHQ
Applied Programming Dept ATTENTION Mr. Robert Bemer

FROM St. Paul

SUBJECT AND OR REFERENCE: Minnesota Association for Digital Computer Applications and Programming

DATE 2/3/58

MSP

I would like to know if you could be available during the latter part of February to address a meeting of the Minnesota Association for Digital Computer Applications and Programming in Minneapolis. This is a new organization of approximately 100 people in the Twin Cities area who are interested in programming techniques and computer applications.

At our original meeting, Grace Hopper from Remington Rand addressed us on the topic of computer languages and spoke in particular of Remington's compilers with very slight mention of our own. For this reason, we would like to present our computer languages in a favorable light, giving them a little publicity during some future meeting.

Dr. Marvin Stein, who is President of this organization, has suggested that you might be the ideal person to present this "rebuttal." We would be very pleased if you could find the time this month to come to Minneapolis, but if this is not possible, I would certainly like to know when you could be in this area.

BOTH SC. EQMM.

RAN
R. A. Zillgitt
Applied Science Representative

RAZ/rk

cc: Mr. Ray N. Hockenberry, Manager, Speakers Bureau, Dept. of Information

CAP 23) 9AM EVR – YIP – 160 MSP (CST)
25) 12 NOON LGA – YIP – 405 MSP
NO 1) 820 AM IDL – YIP – 145 MSP 8

22) 810 AM MSP – YIP – 130 IDL
26) 210 PM MSP – YIP – 725 EVR
2) 430 AM MSP – 716 IDL 8

19 FEB

20 FEB.

THINK

CANNING, SISSON AND ASSOCIATES

Automatic Data Processing and Production Control Systems

Telephone BRADSHAW 2-8425

1140 SOUTH ROBERTSON BOULEVARD
LOS ANGELES 35, CALIFORNIA

June 3, 1958

Mr. Robert W. Bemer
International Business Machines Corp.
590 Madison Avenue
New York, N. Y.

Dear Bob:

Let me compliment you on your fine and successful effort in cataloging automatic programming systems. I have been preparing a comment on the same subject for Data Processing Digest and your information has been a great help.

I wonder if you could find time to classify these automatic programs in two ways. First, as to whether they are designed principally for engineering-scientific applications or for commercial and industrial applications. Secondly, I wonder if you could class them (especially those for business applications), in the following way:

1. Low Level Assembly Programs. These are programs which permit relative addressing and translate almost machine-like codes into the exact machine language. They might perform conversion of letters to numeric codes and decimal to binary.

2. High Level Assembly Programs. These permit the use of symbolic addressing and go through the process of forming a directory to insure the proper assignment of machine addresses throughout the code. They might include a minor use of subroutines.

3. Compilers. Compilers permit the use of macro-instructions or pseudo-instructions. In order to produce the machine code, they incorporate both generators and subroutine libraries.

4. Super-compilers. Super-compilers (my own terminology) include advanced automatic programs; such as the flow-matic, which permit the automatic preparation of codes essentially from the flow-diagram stage.

5. True automatic programs which permit the user to express the problem in the language (if there is one) familiar to him; with the computer performing all of the remaining sequencing (including branching), location of data (including file structure), and compiling.

Even an approximate classification of the programs on your list into these catagories would be appreciated.

Incidentally, if I can be of any further assistance in the preparation of the other article, do not hesitate to write.

Cordially,

Roger L. Sisson

TABLES OF CONTENTS

PROCEEDINGS OF COMPUTER APPLICATIONS SYMPOSIA

FIFTH ANNUAL SYMPOSIUM
October 29-30, 1958

FOURTH ANNUAL SYMPOSIUM
October 24-25, 1957

THIRD ANNUAL SYMPOSIUM
October 9-10, 1956

NOTE: The Proceedings of the First and Second Annual Symposia (1954 and 1955) are out of print.

THE FOURTH
COMPUTER APPLICATIONS SYMPOSIUM

Users of digital computers are continually expanding the frontier of applications into new and complex areas of calculation and data processing. The program of this symposium constitutes a cross section of newer applications and techniques. It illustrates the ever present diversity among both applications and computing machines at industrial and governmental installations. Sharing of computer facilities by management and research interests—to the advantage of both—characterizes many contemporary installations and has its counterpart in the dual conference structure.

Exchange of ideas and experiences among users representing both business and science greatly enlarges our field of vision and promotes more effective use of computers. Armour Research Foundation, one of the world's largest independent research organizations, provides opportunity for such exchange through sponsorship of the Computer Applications Symposium. The Foundation, itself well established in the fields of computer development, operations research, and computing ser[illegible], invites your participation in the Fourth Sy[illegible]sium, confident that it will produce substantial and lasting benefits.

Sessions will stress new applications, use of new computers and accessories, and developments in automatic programming. Round table discussions will follow each day's presentations, emphasizing solutions to planning and operating problems common to the organizations represented.

Symposium Committee
A. K. Hawkes
F. C. Bock
J. J. Kowal

the fourth

computer applications symposium

october 24 & 25, 1957

at the morrison hotel
cotillion room
clark & madison streets
chicago 1, illinois

sponsored by
armour research foundation
of illinois institute of technology

Thursday, October 24, 1957

BUSINESS AND MANAGEMENT APPLICATIONS
COTILLION ROOM, MORRISON HOTEL

Session Chairman: Paul Kircher
Professor of Business Administration
University of California at Los Angeles

8:00 REGISTRATION

9:00 WELCOME
C. E. Watson
Assistant Director
Armour Research Foundation

9:10 **An Extensive Hospital and Surgical Insurance Record-Keeping System (Datamatic 1000)**
R. J. Koch
Director, Office Management Division
Michigan Hospital Service

9:50 **A Central Computer Installation as a Part of an Air Line Reservation System (Univac File-Computer)**
R. A. McAvoy
Superintendent of Communications—Data Processing
Eastern Air Lines

10:30 RECESS

11:00 **Fitting a Computer to an Inventory Control Problem (IBM 705)**
O. A. Kral
Department Head, Applied Mathematics and Statistics
Minnesota Mining and Manufacturing Company

11:40 RECESS

12:20 LUNCHEON, COTILLION ROOM
The Problems of Planning New Metropolitan Transportation Facilities and Some Computer Applications
J. D. Carroll, Jr.
Director
Chicago Area Transportation Study

1:50 **Data Processing Tasks for the 1960 Census (Univac Scientific)**
D. H. Heiser
Chief, Electronic Systems Branch
Bureau of the Census, Department of Commerce

2:30 **The Handling of Retail Store Requisitions from a General Warehouse (Datatron)**
M. J. Stoughton
Computer Engineer
Sears, Roebuck and Company

3:10 RECESS

3:40 **Automatic Programming for Business Applications**
G. M. Hopper
Director of Automatic Programming
Remington Rand Univac Division
Sperry Rand Corporation

4:20 **Round-Table Discussion**
Session Chairmen and Speakers

Friday, October 25, 1957

ENGINEERING AND RESEARCH APPLICATIONS
COTILLION ROOM, MORRISON HOTEL

Session Chairman: Preston C. Hammer
Director
Numerical Analysis Laboratory
University of Wisconsin

8:00 REGISTRATION

9:00 WELCOME
C. E. Barthel
Assistant Director
Armour Research Foundation

9:10 **Digital Simulation of Active Air Defense Systems (Univac Scientific)**
R. P. Rich
Supervisor of the Computer Center
Applied Physics Laboratory
Johns Hopkins University

9:50 **Statistical Calculations in Product Development Research (LGP-30)**
E. B. Gasser
Assistant Director of Research
The Toni Company

10:30 RECESS

11:00 **Progress in Computer Application to Electrical Machine and System Design (IBM 704 and 709)**
E. L. Harder
Director of the Analytical Department
Westinghouse Electric Corporation

11:40 RECESS

12:20 LUNCHEON, COTILLION ROOM
How Lazy Can You Get?
A. L. Samuel
Research Advisor
International Business Machines Corporation

1:50 **The Solution of Certain Problems Occurring in the Study of Fluid Flow (IBM 650)**
L. U. Albers
Chief, Machine Computing Branch
Lewis Flight Propulsion Laboratory
National Advisory Committee for Aeronautics

2:30 **A Dual Use Digital Computer (General Purpose or DDA) for Dynamic System Analysis (G-15D)**
E. H. Clamons and R. D. Adams
Assistant Manager Senior Scientist
of Systems Analysis
Mechanical Division, General Mills, Inc.

3:10 RECESS

3:40 **The Status of Automatic Programming for Scientific Problems**
R. W. Bemer
Manager of Programming Systems
International Business Machines Corporation

4:20 **Round-Table Discussion**
Session Chairmen and Speakers

ARMOUR RESEARCH FOUNDATION OF ILLINOIS INSTITUTE OF TECHNOLOGY
TECHNOLOGY CENTER · 10 WEST 35TH STREET · CHICAGO 16, ILLINOIS · CAlumet 5-9600

ELECTRICAL ENGINEERING
RESEARCH DEPARTMENT
3440 SOUTH STATE STREET

November 7, 1957

Mr. R. W. Bemer
Manager of Programming Systems
International Business Machines Corporation
590 Madison Avenue
New York 22, New York

Dear Mr. Bemer:

On behalf of the conferees and the Foundation, I want to thank you for your contribution to the success of our recent Computer Applications Symposium. We enjoyed hearing your presentation and have had many excellent comments regarding it. You may be interested to know that we have received an influx of requests for copies of the Proceedings from persons who were unable to attend and we believe your contribution was a considerable force in provoking this interest.

We hope you will send us a copy of the paper you presented in the near future since we want very much to include it in the Proceedings.

The preparation of the Proceedings is underway and we look forward to having it available within two months. At that time, several copies will be mailed to you.

Dr. Bock, Mr. Kantner, and I were pleased to make your acquaintance at the Symposium and we hope we shall see you again soon.

Very truly yours,

A. K. Hawkes

A. K. Hawkes, Assistant Supervisor
Mathematical Services Section

AKH:mm

Enclosed are two complimentary copies of the Proceedings of the Fourth Computer Applications Symposium. Your contribution to the Symposium and your cooperation in making early publication of the Proceedings possible are greatly appreciated.

Sincerely yours,

F. C. Bock

F. C. Bock, Operations Analyst
Mathematical Services Section

24 MAR 58

REL

PROGRESS REPORT
on
AUTOMATIC PROGRAMMING SYSTEMS
for
IBM DATA PROCESSING MACHINES

September 12, 1957
9:00 a.m. - 4:00 p.m.

Morning Session - 9:00 a.m.

Introduction

✓ Mr. John McPherson, Vice President,
Programming Research

Automatic Programming Systems for the IBM 650

✓ Mr. Dave Hemmes,
Programming Systems Group

Automatic Programming Systems for the IBM 705

✓ Mr. Roy Goldfinger,
Programming Systems Group

Procedure Synthesis

Mr. William McClelland, Regional Manager,
Applied Science Department

Automatic Programming Systems for the IBM 704

✓ Mr. Dave Hemmes,
Programming Systems Group

Afternoon Session - 1:30 p.m.

Automatic Programming Systems for the IBM 709

✓ Mr. Robert Bemer, Manager,
Programming Systems Group

Report on Activities at Computing Centers

Mr. William Orchard-Hays,
Council for Economic and Industry Research

Prof. John McCarthy, M.I.T.
Computation Center

Mr. Joseph Wegstein,
National Bureau of Standards

The Vanguard 704 Computing Center, 615 Pennsylvania Ave., N. W. will be open at the close of the sessions.

FRIDAY, JANUARY 25th

10:15 a.m. Omnicode—A Common Language Programming System

Omnicode is being used to assemble programs for the IBM 650. Assembly is performed on the IBM 702. Mr. McGee will describe how Omnicode causes the 650 to simulate a pseudo-machine and how it will be expanded to do the same for the 702 in the future.

CARR

Russell C. McGee
Automatic Programming
General Electric Company

11:00 a.m. INTERMISSION

11:30 a.m. A Matrix Compiler for Univac

A matrix mathematical compiler system is being developed by the Franklin Institute staff. Mr. McGinn will review some of the difficulties encountered during this development, and describe what might be expected of the system.

Laurence C. McGinn
Analysis Section
Franklin Institute Laboratories

12:30 p.m. LUNCHEON
Franklin Memorial Hall

2:00 p.m. A Mathematical Language Compiler

Dr. Perlis will discuss a compiler for scientific and engineering problems that was developed for the Datatron and the IBM 650, two medium-sized digital computers. He will give examples of compiled programs, with pertinent programming and compiling times.

HOPPER

Alan J. Perlis
Computation Center
Carnegie Institute of Technology

2:45 p.m. A Mechanized Approach to Automatic Coding

Automatic coding techniques should be practiced at the machine design stage. Dr. Yowell will discuss how the Class 304 EDP System mechanizes automatic coding.

E. C. Yowell
National Cash Register Company

3:30 p.m. INTERMISSION

4:00 p.m. TOUR No. 1

Franklin Institute Univac Computing Center, A-C Network Calculator, and analog computing laboratory.

(or)

TOUR No. 2

Army Signal Supply Agency, IBM 705

5:00 p.m. Adjournment

HOTEL INFORMATION

THE WARWICK

Single rooms	$10, 11, 12, 13, 14
Twin rooms	$14, 15, 16, 17, 18
Suites	$28, 30, 32, 35.

Locust Street at 17th, Phila. 3, Pa.
John P. Schlicher, Office Manager

THE ROBERT MORRIS

Single rooms	$4.75, 5, 5.50, 6
Twin rooms	$8.50, 9.

Arch Street at 17th, Phila. 3, Pa.
Elmer W. Verdier, Assistant Manager

The Franklin Institute

invites you to a symposium on

AUTOMATIC CODING

A symposium that will explore new techniques, and new applications of automatic coding for digital computers in industry and science.

JANUARY 24th and 25th, 1957

THE FRANKLIN INSTITUTE

Benjamin Franklin Parkway at 20th Street

Philadelphia 3, Pennsylvania

Registration Fee: $35

Please try to register in advance

Through advances in automatic coding—or automatic programming, if you prefer, digital computers are being made to do many chores that formerly required the efforts of highly trained technical people. In business, industry, science and engineering, automatic coding will bring digital computer systems to maximum efficiency and usefulness; to what extent this has been achieved—and what remains to be done—will be the substance of this symposium.

PROGRAM

THURSDAY, JANUARY 24th

9:00 a.m. REGISTRATION
Planetarium Lobby

10:00 a.m. WELCOME
Dr. John S. Burlew, Executive Vice President of the Institute.

10:15 a.m. Automatic Coding Experience at the General Electric Company's Univac Installation in Louisville

Mr. Peterson will describe how, to reduce coding time and increase operating efficiency, GE's Louisville staff has used Holt and Turanski's Generalized Programming, Hopper's A-2 Compiler, sorting and merging generators and standardization of programming conventions.

Richard M. Peterson
Major Appliance Division,
General Electric Company

11:00 a.m. INTERMISSION

11:30 a.m. Debugging Automatic Coding

Automatic coding systems sometimes deprive the programmer of detailed knowledge of the computer-coded programs they produce and thus make it difficult to "de-bug" the program. Mr. Katz will discuss how Univac's Automatic Programming Development group is overcoming this disadvantage.

Charles Katz
Remington Rand Univac Division,
Sperry Rand Corporation

12:30 p.m. LUNCHEON
Franklin Memorial Hall

2:00 p.m. Print 1—An Automatic Coding System for the IBM 705

Mr. Bemer will show how this operative system allows the IBM 705 to serve interchangeably as a commercial and scientific computer. Stress has been placed on ease of input-output format control; facilities for supervisory operation are included.

Robert W. Bemer
International Business Machines Corporation

2:45 p.m. The Procedure Translator —A System of Automatic Programming

Mr. Kinzler will outline the areas in which the Translator will find application, and its expected advantages and disadvantages. Mr. Moskowitz will describe the basic ideas and the working operations of the Translator.

Henry H. Kinzler
and Perry M. Moskowitz
Metropolitan Life Insurance Company

3:30 p.m. INTERMISSION

4:00 p.m. TOUR No. 1
Franklin Institute Univac Computing Center, A-C Network Calculator and analog computing laboratory.

(or)

TOUR No. 2
Army Signal Supply Agency, IBM 705 computer.

6:00 p.m. RECEPTION
Refreshments will be served in Franklin Hall Foyer.

6:45 p.m. DINNER
Franklin Memorial Hall

NOTES

Technical sessions will be held in the Institute's Lecture Hall. Guests will find it most convenient to enter by way of the Winter Street door, between 20th and 21st Streets.

Following each paper, time has been set aside for comment and questions from the floor.

Proceedings of the Symposium will be published as a Monograph under auspices of the Journal of The Franklin Institute, and a copy will be sent to each registrant, free of charge. Other copies will be available for purchase at $3.00 per copy.

Reservations for the Symposium should be made in advance, if possible. Registration fee should accompany the reservation request. The fee covers attendance at technical sessions, tours, luncheons, dinner and reception, and a copy of the Proceedings.

If you wish further information, write:

Automatic Coding Symposium
The Franklin Institute
20th and Parkway
Philadelphia 3, Penna.

or 'phone: LO 4-3600, Extension 282. Program Chairman: Mr. Donald B. Houghton, Chief, Analysis Section, Electrical Engineering Division, Franklin Institute Laboratories.