UNCLASSIFIED

AD 401 452

Reproduced
by the

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION, ALEXANDRIA, VIRGINIA

UNCLASSIFIED

TM-(L)-993

TECHNICAL MEMORANDUM

(TM Series)

ASTIA AVAILABILITY NOTICE

Qualified requesters may obtain copies of this report from ASTIA.

This document was produced by SDC in performance of contract AF 19(628)-1648, Space Systems Division Program, for Space Systems Division, AFSC.

160-A Reference Manual

by

Reva P. Bennett
Eugene J. Rosenberg

8 February 1963

Approved

G. L. Myers

SYSTEM
DEVELOPMENT
CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA

The views, conclusions or recommendations expressed in this document do not necessarily reflect the official views or policies of agencies of the United States Government.

Permission to quote from this document or to reproduce it, wholly or in part, should be obtained in advance from the System Development Corporation.

Although this document contains no classified information it has not been cleared for open publication by the Department of Defense. Open publication, wholly or in part, is prohibited without the prior approval of the System Development Corporation.

A-1159

INTRODUCTION

This publication lists and describes all 160-A computer routines available in the Computer Program Development Center (CPDC), and consists of the following six sections:

Section 1.0 - Alphabetical Index	This section is an alphabetical index for all available 160-A routines. It lists identification codes and either locates descriptions appearing in this document, or as in a few instances, references descriptions available in other documents.
Section 2.0 - Subject Classification Code Index	This section is a subject index for all available 160-A routines.
Section 3.0 - Abstracts	This section contains abstracts of all available 160-A routines.
Section 4.0 - Group I Routines	This section contains descriptions of routines found to be more efficient for CPDC use; accordingly these routines are modified and up-dated more often.
Section 5.0 - Group II Routines	This section contains descriptions of routines that have not been employed in the CPDC. Some of these Group II routines are similar to those found in Group I, and other 160-A users may find some of these routines better suited for their applications.
Section 6.0 - Appendix	This section contains the related references to the contents of the manual.

This document provides a complete and current reference manual for all 160-A programming routines available in the CPDC. Use of this document will effect better dissemination of checked-out routines and will reduce redundant programming efforts by 160-A users. It also serves as training aid for new CPDC personnel. In an effort to maintain a realistic and current document, change pages will be published periodically to reflect modifications, additions and/or deletions.

NOTE: Paper tapes are available in the CPDC for each routine listed in this document.

Although source data evolved from SWAP and the CPDC, the majority of the routines are described per SWAP standards. In a few instances, the responsible corporation and/or programmer is not listed, as the information was not available. If upon using this document, you find that you know the answer to any of these voids, please transmit this information to the CPDC.

ALPHABETICAL INDEX

Title	Ident.	Page
ALNUP	Z3.00	4-240
Alphanumeric and Special Character Demonstration Plot	F8.02	4-90
Binary Card-Binary Tape Verify Routine	AF6.06	4-82
Binary Coded Decimal to Binary	H1.01	4-94
Binary Loader for FPP-33 Input	A4.11A	4-7
Binary to Four Bit Decimal Conversion	H1.00	4-93
Bi-Octal Dump	AF3.12	4-67
Bi-Octal Dump 2	F3.06	5-8
Card to Magnetic Tape (167 to 163/164)	F6.03	4-79
Card to Magnetic Tape Routine (160-A)	F6.05	5-10
Card Read Routine (167)	AE2.01	4-34
Change Magnetic Tape Record Length Binary Tapes	F6.04	4-80
Composite Beams	AK4.01	4-197
Conflex	AH1.03	4-100
Core Dump (1612)	AF3.11	4-66
Duplicate Paper Tape	AF1.00	4-37
Equation Solver	AJ1.01	4-152
Flex Load and Flex Verify	AF2.00	4-49
Flex Tape to Magnetic Tape Converter	F6.01	4-75
Flexowriter Tape to Anelex	F4.07	4-70
Floating Bi-Octal Punch and Floating Bi-Octal Verify	AF3.00	4-50
Floating Resident Service Library	AD1.01	4-9
FORTRAN Cards to Flex Tape	AF7.00	4-87
Fractional Arithmetic	I1.04	4-130
General Binary to BCD, BCD to Binary Conversion	AH1.04	4-104
Horizontal Adjustment	AK5.01	4-205
INTERFOR	AB1.03A	**
INTERFOR Arcsine, Arcossine Subroutine	AB1.03-3	**
INTER Assembly Program (FLAP)	AA1.02A	
INTERFOR Exponential Subroutine	AB1.03-5	**
INTERFOR Log to the Base 2 Subroutine	AB1.03-7	**
INTERFOR Sine Cosine Subroutine	AB1.03-4	**
INTERFOR Square Root Subroutine	AB1.03-6	**
INTERFOR Subroutine, Plot	AB1.03-9	**
INTERFOR Subroutines, TRIG Routines	AB1.03-8	**
Integer Divide	I1.03	4-129
Linear Program I	J1.00	4-143
Lines and Circles	AK1.03	4-172
Listable Octal Dump	F3.02	5-7
Listable Symbolic Dump (1 word per Flex line)	F3.04A	4-56

**See "INTERFOR/Reference Manual"

Title	Ident.	Page
Listable Symbolic Dump (4 words per Flex line)	F3.03	4-55
Magnetic Tape Bootstrap (160-A)	AD1.02	4-16
Magnetic Tape to Flex Tape	A3.02	4-5
Matrix Inversion	I1.06	4-135
Memory Dump	F3.01	4-53
Mortgage Amortization	AZ3.03	4-243
Multiple Precision Package	B1.02	5-1
Nine Bit Quick Multiply	I1.02	4-128
Nine Bit Quick Sine	I1.05	4-133
Octal Dump	AF3.09	4-62
OSAS-A Binary Loader - Assembly Mode Magnetic Tape	AA4.03	**
OSAS-A Binary Loader - Card	AA4.02	**
OSAS-A Binary Loader - Character Mode Magnetic Tape	AA4.04	**
OSAS-A Binary Loader - Paper Tape	AA4.01	**
OSAS-A Master Paper Tape Edit	AA2.02	**
OSAS Symbolic Flex Tape to Cards	A3.03	4-6
Paper Tape Edit	A2.01A	4-1
Paper Tape Verify	F1.03	5-6
Peripheral Integrated Utility System	AD2.03	4-28
Plot from Magnetic Tape 163/164 to 165 Plotter	AF8.01	4-88
Program Structure	AI1.07	4-138
Proper Fraction for Calculating Percentages	I1.00 ●	4-126
Punched Paper Tape Duplicator (short tapes)	F1.00	4-35
Punched Paper Tape Duplicator (long tapes)	F1.01	4-38
Random Number Generator (12 bit)	Z1.02	4-238
SICOM	AB1.04A	***
SICOM 167 Data Input ●	AB1.10B	***
SICOM Double Precision Translator	AB1.08A	***
SICOM Double Precision Translator Subroutine	AB1.09A	***
SICOM Graph Plotter Subroutine	AB1.11A	***
SICOM Single Precision Translator	AB1.06A	***
SICOM Single Precision Translator Subroutine	AB1.07A	***
SICOM Standard Mathematical Subroutines	AB1.04-2	***
SICOM Straight Line Plot Routine	AB1.12	***
Simultaneous Card-to-Tape, Tape-to-Print	AD2.02	4-19
Single Precision Divide	I1.01	4-127
Single Precision Fractional Square Root	Z1.00	4-237
Single Profile	AK2.01	4-184
Soil Consolidation	AK6.01	4-214
Sort 3X	H2.01	4-108
Spiralled Way Alignment	AK1.01	4-155

**See "OSAS-A/The 160-A Assembly System."
***See "SICOM Manual, Scientific and Commercial Programming System for Control Data 160-A Computer."

Title	Ident.	Page
Statistical Program I	L1.01	4-227
Statistical Program II	L1.02	4-228
Statistical Program III	L1.03	4-230
Statistical Program IV	L1.04	4-231
Statistical Program V	L1.05	4-232
Statistical Program VI	L1.06	4-233
Statistical Program VII	L1.07	4-235
Symbolic Assembly System (OSAS-A)	AA1.01C	**
Symbolic Flexowriter Dump	AF3.08	4-60
Symbolic Printer Dump	AF3.07	4-57
Symbolic Typewriter Dump	AF3.10	4-63
Tape Compare	AF1.06	4-47
Tape Copy/Compare	F1.05	4-43
Tape Leader Preparation	F1.02	4-41
Tape-to-Card Punching Routine	F5.02	4-73
TRACK	G1.00	4-92
Trajectory Calculation	AZ3.02	4-241
Transformation of Coordinates	AK1.04	4-179
Traverse Adjustment	AK1.02	4-164
160-A Output	AH1.02	4-96
1607 to 160 to 1612 Routine	F4.06	4-68
1609 to Magnetic Tape (1609-164)	F6.02	4-77

**See "OSAS-A/The 160-A Assembly System."

SUBJECT CLASSIFICATION CODE INDEX

A. Assembly and Related Routines

- A1 Assembly Programs
- A2 Symbolic Code Editors
- A3 Peripheral Conversion Routines
- A4 Loaders

B. Interpretive Systems

- B1 Scientific Computations
- B2 Data Processing

C. Compilers

- C1 Scientific Computations
- C2 Data Processing

D. Service Routine Convenience Packages

- D1 Resident Service Packages
- D2 Special Purpose Packages

E. Input-Output Control Routines

- E1 Magnetic Tape
- E2 Card Equipment
- E3 Printers
- E4 Typewriters
- E5 Plotters

F. Service Routines

- F1 Duplicators and Related Routines
- F2 Memory Loaders
- F3 Memory Dumps
- F4 Data Transfer - Printer Output
- F5 Data Transfer - Card Output
- F6 Data Transfer - Magnetic Tape Output
- F7 Data Transfer - Paper Tape Output
- F8 Data Transfer - Plotter Output

SUBJECT CLASSIFICATION CODE INDEX

G. Program Testing Aids

- G1 Trace Routines
- G2 Dynamic Analysis Routines

H. Information Processing Routines

- H1 Conversions
- H2 Sorts and Merges

I. Arithmetic and Elementary Function Routines

- I1 Fixed Point
- I2 Floating Point

J. Linear Programming

K. Civil Engineering

- K1 Surveying
- K2 Highway Engineering
- K3 Hydraulics Engineering
- K4 Structural Engineering
- K5 Photogrammetry
- K6 Soils

L. Statistical Analysis and Probability

Z. Demonstration Routines

- Z1 Mathematical
- Z2 Games
- Z3 Miscellaneous

AA1.01C SYMBOLIC ASSEMBLY SYSTEM (OSAS-A)
ACCEPTS 160-A INSTRUCTIONS IN SYMBOLIC FORM, ASSIGNS ABSOLUTE LOCATIONS TO THE SYMBOLIC ADDRESSES, PERFORMS THE NECESSARY WORK OF RELATIVE ADDRESSING TO PROVIDE AN ABSOLUTE MACHINE CODE FOR INPUT TO THE 160-A COMPUTER. INPUTS CAN BE PAPER TAPE, MAGNETIC TAPE, OR CARDS. OUTPUT IS PAPER TAPE, MAGNETIC TAPE, CARDS, OR PRINTER.

AA1.02A INTERFOR ASSEMBLY PROGRAM (FLAP)
ASSEMBLES A PROGRAM, STATED IN SYMBOLIC LANGUAGE, INTO A FORM EXECUTABLE BY INTERFOR. OUTPUT IS AN ASSEMBLY LISTING TAPE AND A BINARY OBJECT PROGRAM TAPE LOADED BY A4.11. CONFIGURATION - MINIMUM SWAP 160-A, 166 AND 167 INPUT/OUTPUT OPTIONAL.

A2.01A PAPER TAPE EDIT
CHANGES SYMBOLIC PAPER TAPES PREPARED FOR ASSEMBLY BY FLAP OR OSAS. CHANGES MAY BE IN THE FORM OF REPLACEMENTS, INSERTIONS, OR DELETIONS OF COMPLETE LINES.

AA2.02 OSAS-A MASTER PAPER TAPE EDIT
CHANGES AN OSAS-A MASTER PAPER TAPE BY ADDING TO, DELETING AND/OR REPLACING EXISTING ROUTINES IN THE SECOND RECORD.

A3.02 MAGNETIC TAPE TO FLEX TAPE
CONVERT 80 CHARACTER BCD CODED INFORMATION ON MAGNETIC TAPE IN FORMAT OF OSAP CARD INPUT.

A3.03 OSAS SYMBOLIC FLEX TAPE TO CARDS
CONVERTS OSAS SYMBOLIC FLEX TAPES TO HOLLERITH CARDS. ALLOWS OPTIONAL CARD SEQUENCING UP TO FOUR DIGITS.

AA4.01 OSAS-A BINARY LOADER - PAPER TAPE
READS BINARY PROGRAMS ON PAPER TAPE PREPARED BY OSAS-A INTO MEMORY. PROGRAMS ARE RELOCATABLE.

AA4.02 OSAS-A BINARY LOADER - CARD
READS BINARY PROGRAM CARDS PREPARED BY OSAS-A INTO MEMORY. PROGRAMS ARE RELOCATABLE. USES 088 READER.

AA4.03 OSAS-A BINARY LOADER - ASSEMBLY MODE MAGNETIC TAPE
READS BINARY PROGRAM IMAGES PREPARED BY OSAS-A INTO MEMORY. PROGRAMS ARE RELOCATABLE. USES 163 OR 164 TAPE UNITS.

AA4.04 OSAS-A BINARY LOADER - CHARACTER MODE MAGNETIC TAPE
READS BINARY PROGRAM IMAGES PREPARED BY OSAS-A INTO MEMORY. PROGRAMS ARE RELOCATABLE. USES 163 OR 164 TAPE UNITS.

A4.11A INTERFOR BINARY LOADER (FLOADER)
LOADS BINARY OBJECT PROGRAMS PREPARED BY FLAP. THE OBJECT PROGRAM IS RELOCATABLE.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A

B1.02 MULTIPLE PRECISION PACKAGE
PERFORMS ARITHMETIC WITH OPERANDS EXPRESSED TO 6, 12, 18, 24, ...6N BINARY DIGIT PRECISION.
CONFIGURATION-MINIMUM 160 COMPUTER

AB1.03A INTERFOR
AN INTERPRETIVE SYSTEM PROVIDING SYMBOLIC AND NUMERIC PROGRAMMING, 6 INDEX REGISTERS AND 33-BIT FLOATING POINT ARITHMETIC. THE COMMANDS ARE A SUBSET OF THE CONTROL DATA 1604 COMMANDS. PROGRAMS MAY BE WRITTEN FOR INTERFOR WHICH ALSO RUN ON THE 1604.
CONFIGURATION - MINIMUM 160-A COMPUTER.

AB1.03-2 INTERFOR SINE COSINE SUBROUTINE
GIVEN X, COMPUTES THE SINE X OR COS X (WHERE X IS IN RADIANS).

AB1.03-3 INTERFOR ARCSINE, ARCCOSINE SUBROUTINE
GIVEN X, FINDS THE ARCCOS OF X OR ARCSINE OF X.

AB1.03-5 INTERFOR EXPONENTIAL SUBROUTINE
CALCULATES 2 TO THE X POWER, E TO THE X POWER OR 10 TO THE X POWER.

AB1.03-6 INTERFOR SQUARE ROOT SUBROUTINE
GIVEN A NUMBER X IN THE ACCUMULATOR, FINDS THE SQUARE ROOT OF THE NUMBER BY THE USE OF THE NEWTON ITERATION METHOD AND LEAVES THE SQUARE ROOT IN THE ACCUMULATOR.

AB1.03-7 INTERFOR LOG TO THE BASE 2 SUBROUTINE
GIVEN A FLOATING POINT NUMBER IN A, CALCULATE THE LOG TO THE BASE 2 OF THIS NUMBER.

AB1.03-8 INTERFOR SUBROUTINE, TRIG ROUTINES
COMBINES THE FOLLOWING SUBROUTINES - SINE COSINE, ARCTAN X AND SERIES EXPANSION. SINE COSINE COMPUTES THE SIN X OR COS X WHERE X IS IN RADIANS. ARCTANGENT FINDS THE ARCTANGENT OF X USING THE MACLAURIN SERIES AS GIVEN IN HASTINGS, PAGE 137. X IS IN RADIANS.

AB1.03-9 INTERFOR SUBROUTINE, PLOT
PLOTS RESULTS ON THE ON-LINE MODEL 165 PLOTTER IN EITHER OF TWO WAYS - 1. MOVING FROM PREVIOUS PLOT POINT TO PRESENT PLOT POINT IN A STRAIGHT LINE WITH PEN DOWN, AND 2. MOVING FROM PREVIOUS PLOT POINT TO PRESENT PLOT POINT WITH PEN UP, PLOT A SYMBOL TO REPRESENT DESIRED POINT.
CONFIGURATION - MINIMUM 160-A COMPUTER AND MODEL 165 PLOTTER.

AB1.04A SICOM
A GENERAL PURPOSE INTERPRETIVE SYSTEM USING FLOATING POINT ARITHMETIC. APPLICABLE TO SCIENTIFIC, ENGINEERING AND COMMERCIAL COMPUTATIONS. CONVERTS THE 160-A INTO A DECIMAL, FLOATING POINT MACHINE WITH 10 DECIMAL DIGITS, PLUS EXPONENT WORD LENGTH.
CONFIGURATION - CAN RUN WITH A MINIMUM 160-A.

AB1.04-2 SICOM STANDARD MATHEMATICAL SUBROUTINES
THE FOLLOWING SUBROUTINES ARE INCLUDED - SQUARE ROOT, SIN-COS, LOG X TO BASE 10, 10 TO THE X POWER AND ARC TAN.

B1.05 160 FORTRAN FLOATING POINT ARITHMETIC PACKAGE
PROVIDES A SEPARATE ASSEMBLY OF THE ARITHMETIC PACKAGE USED IN 160 FORTRAN FOR INCORPORATION AS A SUBROUTINE IN PROGRAMS WHICH REQUIRE SUCH FLOATING ARITHMETIC.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A COMPUTER.

AB1.06A SICOM SINGLE PRECISION TRANSLATOR
THE SINGLE PRECISION INTERCOM TRANSLATOR IS A PROGRAM BY WHICH A BENDIX G-15 INTERCOM 1000 SINGLE PRECISION OR INTERCOM 500X SOURCE PROGRAM PREPARED ON THE FLEXOWRITER MAY BE TRANSLATED INTO A CONTROL DATA 160-A SICOM SOURCE PROGRAM TAPE.

AB1.07A SICOM SINGLE PRECISION TRANSLATOR SUBROUTINES
INCLUDED WITH THE TRANSLATOR ARE SICOM SUBROUTINES WHICH ARE SUBSTITUTED FOR THEIR G-15 COUNTERPARTS (I.E., SQUARE ROOT, SIN-COS, ARCTAN, LOG, EXPONENTIAL AND MAGNETIC TAPE). THE INTERCOM FRACTION SELECTOR AND FLEXOWRITER SUBROUTINES ARE REPLACED BY SICOM COMMANDS. THE G-15 INDEX REGISTER UTILIZATION SUBROUTINE IS UNNECESSARY IN SICOM.

AB1.08A SICOM DOUBLE PRECISION TRANSLATOR
THE DOUBLE PRECISION INTERCOM TRANSLATOR IS A PROGRAM BY WHICH A BENDIX G-15 INTERCOM 1000 DOUBLE PRECISION SOURCE PROGRAM PREPARED ON THE FLEXOWRITER MAY BE TRANSLATED INTO A CONTROL DATA 160-A SICOM SOURCE TAPE. THE TRANSLATION WILL, IN MOST CASES, BE READY FOR A PRODUCTION RUN ON THE 160-A UNDER SICOM CONTROL. THE TRANSLATOR IS DESIGNED TO MAKE THE CONVERSION OF ANY WELL-DOCUMENTED INTERCOM DOUBLE PRECISION PROGRAM AS SIMPLE AS POSSIBLE FOR A PROGRAMMER FAMILIAR WITH BOTH SYSTEMS.

AB1.09A SICOM DOUBLE PRECISION TRANSLATOR SUBROUTINES
INCLUDED WITH THE TRANSLATOR ARE SICOM SUBROUTINES WHICH ARE SUBSTITUTED FOR THEIR G-15 COUNTERPARTS. THE INTERCOM FRACTION SELFCTOR AND THE G-15 INTERCOM INDEX REGISTER UTILIZATION SUBROUTINES IS UNNECESSARY IN SICOM.

AB1.10B 167 DATA INPUT FOR SICOM
FIXED POINT INTEGERS AND ALPHANUMERIC DATA MAY BE PUNCHED ON CARDS AND LOADED VIA THE 167 READER. THE SUBROUTINE MUST BE ENTERED EACH TIME A CARD IS READ.
CONFIGURATION - MINIMUM SWAP 160-A, 167 READER.

AB1.11A SICOM GRAPH PLOTTER SUBROUTINE
RELOCATABLE GENERAL PURPOSE PLOTTER SUBROUTINE.

AB1.12 SICOM STRAIGHT LINE PLOT ROUTINE
PLOTS STRAIGHT LINES IN AN X, Y OR 45 DEGREE DIRECTION.

AC1.00 160 FORTRAN-A SYSTEM
PROVIDES A SIMPLE AND EFFICIENT METHOD FOR SCIENTIFIC PROBLEM SOLVING. EMPLOYS A LANGUAGE RESEMBLING THAT OF ORDINARY MATHEMATICS AND COMMON LANGUAGE TO DESCRIBE THE MATHEMATICAL AND LOGICAL PROCEDURE TO BE FOLLOWED TO CALCULATE THE DESIRED RESULTS. THIS LANGUAGE (SOURCE PROGRAM) IS THE INPUT TO FORTRAN WHICH PRODUCES MACHINE INSTRUCTIONS (OBJECT PROGRAM) THAT PERFORM THE DESIRED CALCULATIONS.
CONFIGURATION - MINIMUM SWAP 160-A, OPTIONAL 088 OR 167.
THE SYSTEM CONSISTS OF PHYSICAL TAPES LISTED BELOW.
160 FORTRAN-A COMPILER I (088 SOURCE INPUT)
160 FORTRAN-A COMPILER I (167 SOURCE INPUT)
160 FORTRAN-A COMPILER II
160 FORTRAN-A INTERPRETER

AD1.01 RESIDENT SERVICE PACKAGE 1
EACH OF THE FOLLOWING FUNCTIONS IS OPERATED BY ENTERING THE RESIDENT LIBRARY AT A SPECIFIED POINT - BI-OCTAL PUNCH, BI-OCTAL VERIFY, FLEX TAPE LOAD, FLEX TAPE PUNCH, FLEX TAPE VERIFY, TYPE LOAD, AND TYPE DUMP.
CONFIGURATION - MINIMUM 160-A COMPUTER. TYPEWRITER OPTIONAL.

AD1.02 MAGNETIC TAPE BOOTSTRAP
KEEPS ITSELF AND PROGRAMS OR DATA ON MAGNETIC TAPE TO ELIMINATE PAPER TAPE HANDLING. MAY BE LOADED AND BLOCKS WRITTEN OR READ BY CONSOLE ENTRY OF DATA.
CONFIGURATION-MINIMUM SWAP 160-A AND A 163-2.

AD2.02 SIMULTANEOUS CARD-TO-TAPE, TAPE-TO-PRINT PROGRAM
RUNS A CARD-TO-TAPE OPERATION SIMULTANEOUSLY WITH A TAPE-TO-PRINTER OPERATION. ABOUT 90 PER CENT OF MAXIMUM SPEED IS OBTAINED.
CONFIGURATION - MINIMUM SWAP 160-A, 163-A, 163-2, 1612, 088.

AD2.03 PERIPHERAL INTEGRATED UTILITY SYSTEM (PIUS)
TO PROVIDE A PROGRAM TO CORRELATE AND CONTROL A LIBRARY OF 160-A ROUTINES IN AN EXECUTIVE UTILITY SYSTEM.

AE2.01 167 CARD READ ROUTINE
READS A HOLLERITH OR BINARY CARD AND STORES THE INFORMATION PACKED, (12 BITS PER WORD), OR UNPACKED, SIX BITS IN THE LOWER HALF OF EACH WORD.
CONFIGURATION - MINIMUM SWAP 160-A, 167 READER.

MATION PACKED, (12 BITS PER WORD), OR UNPACKED, SIX BITS
IN THE LOWER HALF OF EACH WORD.
CONFIGURATION - MINIMUM SWAP 160-A, 167 READER.

F1.00 PAPER TAPE DUPLICATOR (TAPDUP)
PRODUCES AND VERIFIES MULTIPLE COPIES OF A GIVEN PUNCHED PAPER TAPE. COPIES UP TO 6395 FRAMES OF TAPE AND PRODUCES THE NUMBER OF COPIES SPECIFIED IN THE A REGISTER.
PUNCHES AN 18 INCH LEADER BETWEEN COPIES.
CONFIGURATION- MINIMUM 160 COMPUTER.

F1.01 PAPER TAPE DUPLICATOR, LONG TAPES (TAPDPL)
REPRODUCES AND VERIFIES PAPER TAPES OF MORE THAN 6395 FRAMES BY PLACING THE IMAGE ON MAGNETIC TAPE. REQUIRES 212 LOCATIONS BEGINNING AT LOCATION 0050.
CONFIGURATION- MINIMUM 160 COMPUTER AND 163 OR 164 TAPE UNITS.

F1.02 TAPE LEADER PREPARATION (TAPIDT)
PRODUCES READABLE CHARACTERS ON PAPER TAPE LEADERS FOR IDENTIFYING PROGRAM TAPES.
CONFIGURATION- MINIMUM 160 COMPUTER.

F1.03 PAPER TAPE VERIFY (FLXLOD)
VERIFIES PAPER TAPES BY COMPARING THEM AGAINST CHECK SUMS FORMED BY THE ORIGINAL TAPE. ANY LENGTH TAPE BE VERIFIED.
CONFIGURATION- MINIMUM 160 COMPUTER

F1.05 TAPE COPY - COMPARE
PROVIDES THE FOLLOWING OPTIONS - COPY OR COMPARE MAGNETIC TAPE FILES, BCD OR BINARY MODE, REWIND LOAD AT START, REWIND UNLOAD AT FINISH.
CONFIGURATION-MINIMUM 160-A COMPUTER.

AF1.06 TAPE COMPARE (TACO)
TO COMPARE TWO BCD TAPES TO INSURE SIMILARITY
CONFIGURATION-MINIMUM 160 A, 161 TYPEWRITER,
1612PRINTER (OPTIONAL)

AF2.00 FLEX LOAD AND FLEX VERIFY
READS PAPER TAPES INTO MEMORY THAT WERE PREPARED ON A FLEXOWRITER OR BY THE FLEX DUMP PROGRAM. FLEX VERIFY CHECKS FOR DISCREPANCIES BETWEEN TAPE AND MEMORY.
CONFIGURATION-MINIMUM 160-A COMPUTER.

AF3.00 BI-OCTAL PUNCH AND VERIFY (BIOPUN)
PUNCHES PAPER TAPE AND COMPARES PORTIONS OF MEMORY IN BI-OCTAL FORMAT.
CONFIGURATION- MIMIMUM 160 COMPUTER.

F3.01 MEMORY DUMP TO MAGNETIC TAPE (MEMTPE)
REQUIRES 181 LOCATIONS
CONFIGURATION- MINIMUM 160 COMPUTER AND 163 OR 164 TAPE UNITS.

F3.02 LISTABLE OCTAL DUMP (MEMFLX)
PRODUCES IN FLEXOWRITER CODE A LISTABLE OCTAL DUMP OF A SPECIFIED AREA OF CORE.
CONFIGURATION- MINIMUM 160 COMPUTER.

F3.03 LISTABLE SYMBOLIC DUMP, 4 WORDS TO A LINE (MEMSY4)
PRODUCES IN FLEXOWRITER CODE A LISTABLE SYMBOLIC DUMP OF A SPECIFIED AREA OF CORE, FOUR WORDS PER FLEX LINE.
CONFIGURATION- MINIMUM 160 COMPUTER.

F3.04A LISTABLE SYMBOLIC DUMP, 1 WORD TO A LINE. (MEMSYL)
PRODUCES IN FLEXOWRITER CODE A LISTABLE SYMBOLIC DUMP OF A SPECIFIED AREA OF CORE, ONE WORD PER FLEX LINE.
CONFIGURATION- MINIMUM 160 COMPUTER.

F3.06 BI-OCTAL DUMP (MEMTAP)
DUMPS MEMORY ONTO PAPER TAPE IN BI-OCTAL FORMAT FROM SELECTED ADDRESSES.
CONFIGURATION- MINIMUM 160 OR 160A COMPUTER.

AF3.07 SYMBOLIC PRINTER DUMP
DUMPS MEMORY IN SYMBOLIC FORM ONTO THE 1612 PRINTER.
CONFIGURATION-MINIMUM 160-A COMPUTER, 1612 PRINTER.

AF3.08 SYMBOLIC FLEXOWRITER DUMP
DUMPS MEMORY IN SYMBOLIC FORM ONTO PUNCHED PAPER TAPE FOR LISTING ON FLEXOWRITER.
CONFIGURATION-MINIMUM 160-A.

AF3.09 OCTAL DUMP
PROVIDES AN OCTAL DUMP OF ONE BANK OF 160-A MEMORY FROM A GIVEN STARTING ADDRESS TO THE END OF THAT BANK ONTO THE 1612 PRINTER. THE PROGRAM IS RELOCATABLE.

CONFIGURATION-MINIMUM 160-A COMPUTER, 1612 PRINTER.

AF3.10 SYMBOLIC TYPEWRITER DUMP
PROVIDES A SYMBOLIC DUMP ONTO THE 161 TYPEWRITER.
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

AF3.11 160-A CORE DUMP TO 1612 (1612 DUMP)
TO ALLOW A LISTABLE DUMP OF THE MACHINE CODES AND CONSTANTS STORED IN 160A AND 169 ONTO THE 1612 PRINTER.
CONFIGURATION-MINIMUM 160 A, 161 TYPEWRITER, 1612 PRINTER

AF3.12 BIOCTAL DUMP (BIDU)
PUNCHES A GIVEN AREA OF A SELECTED BANK IN BIOCTAL FORMAT FOR MACHINE LOADING.
CONFIGURATION-MINIMUM 160 A.

F4.06 1607 TO 1612 PRINT ROUTINE
READS BLOCKED OR UNBLOCKED MAGNETIC TAPES AND LISTS THEM ON THE CONTROL DATA 1612 PRINTER AT 1000 LINES A MINUTE. LIST CAN BE UNDER PROGRAM CONTROL OR UNDER CONTROL OF 1612 MONITOR CHANNEL 1-6.
CONFIGURATION-MINIMUM 160 OR 160-A COMPUTER, 1607, AND 1612.

F4.07 FLEX TAPE TO 1612
READS FLEX-CODED PAPER TAPE, CONVERTS THE FLEX CODES INTO BCD, PRINTS THE RESULTS ON THE 1612.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A AND 1612.

F5.02 163/164 TAPE TO CARDS
READS MAGNETIC TAPE IN BCD OR BINARY MODE AND PUNCHES CORRESPONDING HOLLERITH OR BINARY CARDS. WRITTEN PRIMARILY TO PUNCH OUTPUT FROM THE 1604 CO-OP MONITOR.
CONFIGURATION-MINIMUM SWAP 160 OR 160-A, 163 OR 164 TAPE UNIT, 1610-523 CARD PUNCH SYSTEM.

F6.01 FLEX TAPE TO MAGNETIC TAPE CONVERTER (FLXTPE)
REPRODUCES A COPY OF A FLEXOWRITER TAPE ON MAGNETIC TAPE IN A FORM SUITABLE FOR LISTING.
CONFIGURATION- MINIMUM 160 COMPUTER AND 163 OR 164 TAPE UNITS.

F6.02 1609 CARD TO MAGNETIC TAPE
READS BCD CARDS, CHECKS VALIDITY AND WRITES BCD CARD IMAGES

ONTO MAGNETIC TAPE. REQUIRES LOCATIONS 6600 THROUGH 0000 AND LOCATIONS 0070 THROUGH 0076.
CONFIGURATION - MINIMUM 160 OR 160-A, 1609 ADAPTER, 164 TAPE AND A 521 CARD READ PUNCH.

F6.03 167 CARD TO TAPE
CONVERTS DATA FROM CARDS TO MAGNETIC TAPE RECORDS, ONE CARD PER RECORD. HANDLES BINARY OR BCD CARDS. PRODUCES BINARY OR BCD RECORDS. CARDS MAY BE MIXED.
CONFIGURATION-MINIMUM SWAP 160 OR 160-A, 167 READER AND A 163 OR 164 TAPE UNIT.

F6.04 CHANGE MAGNETIC TAPE RECORD LENGTH BINARY TAPES
WRITES MAGNETIC TAPES OF A DESIGNATED RECORD LENGTH FROM TAPES OF THE SAME OR DIFFERENT RECORD LENGTHS FOR BINARY TAPES ONLY (ODD PARITY).
CONFIGURATION - MINIMUM SWAP 160 OR 160-A, 163-2 OR 164-2 TAPE UNIT.

F6.05 160-A CARD TO MAGNETIC TAPE ROUTINE
ROUTINE ACCEPTS STANDARD 80 COLUMN BCD OR COLUMN BINARY CARDS AND WRITES THEIR CARD IMAGES WITH ''LOOK-AHEAD'' BITS ONTO MAGNETIC TAPE.

AF6.06 160-A BINARY CARD BINARY TAPE VERIFY ROUTINE
ROUTINE COMPARES 80 COLUMN BINARY CARDS WITH THEIR IMAGE ON MAGNETIC TAPE. WRITTEN FOR IBM 533 CARD READ PUNCH.
CONFIGURATION-160-A WITH 163 AND 1610 ADAPTOR FOR IBM CARD READ PUNCH.

AF7.00 FORTRAN CARDS TO FLEX TAPE
READS FORTRAN SOURCE CARDS AND PUNCHES PAPER TAPE ACCEPTABLE TO THE FORTRAN COMPILER.
CONFIGURATION-MINIMUM SWAP 160-A, 167 READER.

AF8.01 PLOT FROM MAGNETIC TAPE 163/164 TO 165 PLOTTER
PLOTS CONSECUTIVE POINTS ON THE 165 PLOTTER AND CONNECTS THEM WITH THE BEST STRAIGHT LINE. POINTS ARE OBTAINED FROM MAGNETIC TAPE, ONE POINT PER TWO-WORD ASSEMBLY-MODE RECORD.
CONFIGURATION - MINIMUM SWAP 160-A COMPUTER, TAPE UNIT 163 OR 164 AND 165 PLOTTER.

F8.02 165 ALPHANUMERIC AND SPECIAL CHARACTER DEMONSTRATION PLOT

BASICALLY A DEMONSTRATION ROUTINE, BUT IT MAY BE USED TO PRODUCE A NEAT LABELLING OF PLOTS ON THE 165 PLOTTER.
CONFIGURATION - MINIMUM 160 OR 160-A SWAP COMPUTER, 165 PLOTTER.

G1.00 TRACK (TRACE1)
TRACES A PROGRAM, PROVIDING FLEX-CODED PAPE TAPE AS OUTPUT. THE BEGINNING AND ENDING ADDRESSES OF CONSECUTIVE INSTRUCTION STRINGS APPEAR AS OUTPUT, ALLOWING FASTER EXECUTION THAN IS POSSIBLE USING A FULL TRACE.
CONFIGURATION- MINIMUM UNMODIFIED 160 COMPUTER.

H1.00 BINARY TO 4 BIT DECIMAL CONVERSION (BINDEC)
CONVERTS A 24 BIT BINARY INTEGER INTO A DECIMAL INTEGER WITH EACH DIGIT IN SUCCESSIVE CELLS.
CONFIGURATION- MINIMUM 160 COMPUTER.

H1.01 BCD TO BINARY CONVERSION (BCDBIN)
CONVERTS A BINARY CODED DECIMAL NUMBER OF UP TO SIX DIGITS TO THE EQUIVALENT BINARY NUMBER IN 22 BIT ARITHMETIC FORMAT. THE BCD NUMBER IS STORED ONE DIGIT PER WORD WITH THE DIGIT AS THE LOW ORDER FOUR BITS OF THE WORD.
CONFIGURATION- MINIMUM 160 COMPUTER AND 163 OR 164 TAPE UNITS.

AH1.02 OUTPUT IMAGE
SETS UP A 120 CHARACTER LINE IMAGE FOR ON-LINE PRINTING OR WRITING ON A BCD OUTPUT TAPE. THE FORMAT IS SPECIFIED IN THE CALLING SEQUENCE.

AH1.03 CONFLEX
CONVERTS THE FLEXOWRITER MODEL 35-4(STANDARD WITH G-15) IN THE 6 LEVEL MODE TO EQUIVALENT CONTROL DATA FLEXO-WRITER CHARACTERS.
CONFIGURATION-MINIMUM SWAP 160-A.

AH1.04 GENERAL BINARY TO BCD/BCD TO BINARY CONVERSION
A CLOSED SUBROUTINE ENTERED BY A RETURN JUMP WITH THE A-REGISTER SPECIFYING THE TYPE OF CONVERSION.
CONFIGURATION-MINIMUM SWAP 160-A.

H2.01 SORT 3X
SORT 3X ACCEPTS INPUT FROM ONE REEL OF MAGNETIC TAPE WHERE THE INFORMATION IS RECORDED WITH ONE ITEM PER BLOCK OF MAGNETIC TAPE.

CONFIGURATION-MINIMUM SWAP 160 OR 160-A COMPUTER AND FOUR 606, 1607, 163, OR 164 TAPE HANDLERS.

I1.00 PECENTAGE CALCULATION, PROPER FRACTION (PERCAL)
DIVISION OF X OVER Y, WHERE X IS ASSUMED TO BE LESS THAN Y. THE ANSWER WILL BE IN THE FORM .XXXX, ADJUSTED BY ROUNDING IN TRUE BINARY
REQUIRES 196 LOCATIONS, INCLUDING 70-77. RELOCATABLE.
CONFIGURATION- MINIMUM 160 COMPUTER.

I1.01 SINGLE PRECISION DIVIDE (DIVFRC)
DIVIDES A POSITIVE 23 BIT FRACTION BY A POSITIVE 11 BIT FRACTION, GIVING A ROUNDED 11 BIT FRACTIONAL QUOTIENT.
CONFIGURATION- MINIMUM 160 COMPUTER.

I1.02 QUICK MULTIPLY, 9 BIT (MULT9B)
MULTIPLIES TWO SIGNED 11 BIT NUMBERS TOGETHER. SIGNED ANSWER IS ACCURATE TO 10 BITS.
CONFIGURATION- MINIMUM 160 COMPUTER.

I1.03 INTEGER DIVIDE (DIVINT)
DIVIDES A POSITIVE 23 BIT INTEGER BY A POSTIVE 11BIT INTEGER, GIVING A 12 BIT QUOTIENT WITH AN 11 BIT REMAINDER.
REPEATED SUBTRACTIONS ARE USED.
CONFIGURATION- MINIMUM 160 COMPUTER

I1.04 FRACTIONAL ARITHMETIC - 22 BIT
ADDS, SUBTRACTS, MULTIPLIES OR DIVIDES POSITIVE OR NEGATIVE BINARY FRACTIONS.
CONFIGURATION-MINIMUM SWAP 160 OR 160-A.

I1.05 NINE BIT QUICK SINE
OBTAINS THE SINE OF ANY ANGLE LESS THAN 90 DEGREES.
ACCURATE TO 9 BITS. USES I1.02.
CONFIGURATION-MINIMUM 160 OR 160-A COMPUTER.

I1.06 MATRIX INVERSION
SUBROUTINE IS WRITTEN IN 160 FORTRAN (160 FORTRAN-A) FOR FINDING THE INVERSE OF A SQUARE MATRIX.

AI1.07 PROGRAM STRUCTURE
DETERMINES ALL THOSE PERMUTATIONS WHICH LEAVE A STRUCTURE (S,R) INVARIANT, AND ALL THOSE PERMUTATIONS

RHO WHICH LEAVE (S,R) SELF-DUAL.
CONFIGURATION-MINIMUM SWAP 160-A.

J1.00 LINEAR PROGRAM I
FORTRAN. USES THE SIMPLEX METHOD. THE INPUT DATA MUST IMPLY A FIRST FEASIBLE SOLUTION.
CONFIGURATION-MINIMUM SWAP 160 OR 160-A.

AJ1.01 EQUATION SOLVER
FORTRAN. SOLVES 30 LINEAR EQUATIONS WITH 30 UNKNOWNS. A MODIFIED GAUSSIAN ELIMINATION METHOD IS USED.
CONFIGURATION-MINIMUM SWAP 160 OR 160-A.

AK1.01 SPIRALLED WAY ALIGNMENT
COMPUTES THE ALIGNMENT AND CURVE DATA FOR A HORIZONTAL ALIGNMENT CONTAINING SPIRALS AND CIRCULAR CURVES IN ANY COMBINATION.
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK1.02 TRAVERSE ADJUSTMENT
FOR THIS PROGRAM, A TRAVERSE IS DEFINED AS A SERIES OF MEASURED DISTANCES AND DIRECTIONS, STARTING FROM A POINT OF KNOWN COORDINATES, AND TERMINATING AT A POINT OF KNOWN COORDINATES. THE PROGRAM PROVIDES FOR THE OUTPUT OF EITHER THE ADJUSTED OR UNADJUSTED TRAVERSE (ACCORDING TO COMPASS RULE).
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK1.03 LINES AND CIRCLE
THIS PROGRAM CONSISTS OF THE FOLLOWING FIVE INDEPENDENT PARTS. THE KNOWN DATA MAY BE LOCATED IN ANY ONE OF FOUR QUADRANTS.
PART 1 - INTERSECTIONS OF TWO LINES
PART 2 - INTERSECTION OF LINE AND CIRCLE
PART 3 - DISTANCE AND BEARING
PART 4 - HEIGHT OF TRIANGLE
PART 5 - TANGENT, ARC, AND SEGMENT
CONFIGURATION - MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK1.04 TRANSFORMATION OF COORDINATES
THIS PROGRAM WILL ROTATE AND TRANSLATE THE COORDINATES OF SERIES OF POINTS WHICH DEFINE AN ENCLOSED AREA OR TRAVERSE, COMPUTE THE BEARINGS AND DISTANCES BETWEEN THESE POINTS, AND FIND THE AREA ENCLOSED.
CONFIGURATION - MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK2.01 SINGLE PROFILE
GIVEN BASIC DATA TO DEFINE THE VERTICAL ALIGNMENT OF A HIGHWAY THE PROGRAM WILL COMPUTE ALL THE REQUIRED DATA AND ELEVATIONS AT EVEN INCREMENTED STATIONS.
CONFIGURATION - MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK4.01 COMPOSITE BEAMS
DESIGNS INTERIOR OR EXTERIOR STEEL BEAMS FOR SIMPLE SPAN HIGHWAY BRIDGES. THE FIRST PART COMPUTES A SOLUTION FOR INTERIOR BEAMS, THE SECOND PART COMPUTES A SOLUTION FOR FASCIA OR EXTERIOR BEAMS. EACH PART MAY BE RUN INDEPENDENTLY.
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK5.01 HORIZONTAL ADJUSTMENT
IT IS DESIRED TO OBTAIN MAP OR GRID COORDINATES OF PHOTOGRAMMETRIC POINTS, NAMELY THE HORIZONTAL PASS POINTS, AS THE RESULTS OF STEREOTRIANGULATION. THE MAP COORDINATES OF HORIZONTAL CONTROL POINTS ARE GIVEN. THE INSTRUMENT COORDINATES OF THE CONTROL POINTS AND THE PASS POINTS ARE OBSERVED FOR A TRIANGULATED STRIP OF CONTROLLED PHOTOGRAPHY.
CONFIGURATION - MINIMUM SWAP 160-A, 161 TYPEWRITER.

AK6.01 SOIL CONSOLIDATION
COMPUTES THE SETTLEMENT OF N LAYERS OF SOIL WITH A MAXIMUM OF FOUR LAYERS AND A SURCHARGE LOAD PLACED IN N NUMBER OF LIFTS WHICH COULD BE TOTALLY OR PARTIALLY REMOVED.
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

L1.01 STATISTICAL PROGRAM 1
PRODUCES THE MEANS AND STANDARD DEVIATIONS FOR TWO SETS OF VARIABLES AND THE CORRELATION BETWEEN CORRESPONDING ENTRIES IN THE TWO LISTS. FORTRAN.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A COMPUTER.

L1.02 STATISTICAL PROGRAM 2
LINEAR REGRESSION ROUTINE WHICH PRODUCES QUANTITIES FOR TWO SETS OF VARIABLES.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A COMPUTER.

L1.03 STATISTICAL PROGRAM 3
TEST OF NORMALITY IN A FREQUENCY DISTRIBUTION OF LARGE SIZE.

CONFIGURATION - MINIMUM SWAP 160-A OR 160 COMPUTER.

L1.04 STATISTICAL PROGRAM 4
PRODUCES THE MEAN AND STANDARD DEVIATION FOR A SINGLE VARIABLE. FORTRAN.
CONFIGURATION - MINIMUM SWAP 160 OR 160-A COMPUTER.

L1.05 STATISTICAL PROGRAM 5
PRODUCES THE BASIC STATISTICAL BUILDING BLOCKS FOR A THREE-DIMENSIONAL SAMPLE-SPACE, I.E., SUMS, SUMS OF SQUARES AND SUMS OF CROSS PRODUCTS. FORTRAN.
CONFIGURATION - MINIMUM SWAP 160 OR 160A COMPUTER.

L1.06 STATISTICAL PROGRAM 6
PRODUCES THE PERCENTILES AT 5 PERCENT INTERVALS AFTER SORTING THE ORIGINAL DATA INTERNALLY. FORTRAN.
CONFIGURATION - MINIMUM SWAP 160 OR 160A COMPUTER.

L1.07 STATISTICAL PROGRAM 7
PRODUCES AND OUTPUTS EITHER THE T-TEST VALUE FOR EQUAL-SIZED SETS OR THE T-TEST VALUE FOR UNEQUAL-SIZED SETS, DEPENDING ON WHETHER THE SETS ARE EQUAL OR UNEQUAL IN LENGTH.
CONFIGURATION - MINIMUM SWAP 160-A OR 160 COMPUTER.

Z1.00 SINGLE PRECISION FRACTIONAL SQUARE ROOT (DEMSQR)
FINDS THE SQUARE ROOT OF A PROPER FRACTION WITH MAXIMUM ERROR OF 2 TO THE MINUS 11.
CONFIGURATION- MINIMUM 160 COMPUTER.

Z1.02 RANDOM NUMBER GENERATOR (12 BIT) (DERMGN)
PRODUCES 12 BIT RANDOM NUMBERS. WILL GENERATE 2,560,000 NUMBERS WITHOUT REPEATING. OUTPUT IS DISPLAYED IN REGISTER A.
CONFIGURATION- MINIMUM 160 COMPUTER.

Z3.00 ALNUP
ROUTINE PUNCHES CHARACTER MESSAGES IN PAPER TAPE THAT ARE LEGIBLE TO AN UNSKILLED OBSERVER.
CONFIGURATION-160 CODED FLEXOWRITER. 160 COMPUTER WITH PAPER TAPE INPUT AND OUTPUT.

AZ3.02 TRAJECTORY CALCULATION
GIVEN THE TARGET RANGE, THE VELOCITY OF THE PROJECTILE, AND ITS ANGLE OF INCLINATION, THE PROGRAM WILL PLOT THE PATH OF THE PROJECTILE ON THE TYPEWRITER.
CONFIGURATION-MINIMUM 160-A AND TYPEWRITER.

AZ3.03 MORTGAGE AMORTIZATION
THE PROGRAM HAS TWO OPTIONS - 1. COMPUTES THE MONTHLY PAYMENT NECESSARY TO AMORTIZE A GIVEN PRINCIPAL AMOUNT AT A GIVEN INTEREST RATE FOR GIVEN DURATION, AND
2. COMPUTES THE DURATION NECESSARY TO AMORTIZE A GIVEN INTEREST RATE WITH A GIVEN MONTHLY PAYMENT.
CONFIGURATION-MINIMUM SWAP 160-A, 161 TYPEWRITER.

A2.

IDENTIFICATION

Title: Paper Tape Edit - Ident. A2.01A
Programmer: Harold C. Schnackel, February 1961

PURPOSE

This program permits changes to be made to symbolic paper tapes prepared for assembly via FLAP, or OSAS. Changes may be in the form of replacements, insertions, or deletions of complete lines relative to the tape to be corrected.

USAGE

1. Operational Procedure

   a. Machine load bioctal program tape at location 0000.

   b. Position correction tape in reader, master clear, and run.

   c. When correction tape has stopped, position the tape to be corrected in the reader, and run. The original tape will read in and the corrected tape will be punched out.

   d. NORMAL STOPS (OCTAL)

      0657: Correction tape has been read in correctly, RUN from here to read tape to be corrected.

      1067: WAI pseudo-op on the original tape has been encountered. If more input is to be processed, RUN. If no more input is to be processed, make the A-Reg. non-zero and RUN. This will complete output punching and indicate if all corrections have been processed.

      1106: END pseudo-op has been encountered or WAI has been previously encountered. The input tape has been edited completely. This is the final normal stop.

3. Space required

   a. Program plus transient storage occupies locations 0000 through 1232.

   b. Input from the correction tape occupies locations 1467 through 7776 as needed.

   c. Locations 1267 through 1466 are reserved for output characters.

7. ERROR STOPS (OCTAL)

0512: There was an error in the edit tape; a line was referenced incorrectly. Continuation will cause the current correction code to be bypassed and the next to be processed.

0572: Computer capacity for holding the corrections has been exceeded. Corrections must be reorganized into two or more passes.

0755: A non-digit character, with the exception of blank, space, and delete code, is present in the additive field of a correction code line. Correction tape must be corrected.

1032: No correction code found - machine error.

1057: An illegal edit code (outside the set d, i, r, z) has been encountered. Restart after correcting correction tape.

1107: Corrections not completed. RUN to punch out correction code not processed.

1207: Final error stop after 1107 has been reached.

10. Input and Output Formats

FLAP and OSAS formats may be used on the tape to be corrected. The format of a correction specification line on the correction tape is as follows:

a. Tab

b. A letter of the set d, i, r, z where
   d = delete. The delete code followed by two minus signs (--) in the address field and n in the additive field will cause n lines of information to be deleted from the symbolic paper tapes.

   i = insert
   r = replace
   z = end of correction tape

c. Tab

d. Location symbol on original tape

e. Tab

f. Pure decimal digit address relative to the location symbol. The

absence of a location symbol means that this numerical quantity is an absolute line count.

g. Carriage Return

The following example illustrates formats of input and output tapes.
Symbolic tape to be corrected:

```
abcd    ldf   qzr
        adn   22
        sbf   qxr
        sti   cntr
        ldf   02
        jfi   02
              next
              subr
next    aod   cntr
        zjf   01
        jpi   xyz
        hlt   00
        end
```

Corrections to be applied:

```
r       abcd  1
add     XYZ
d       --    3
i       abcd  7
aod     stu
zjf     05
d       --    2
d       next  1
d       --    1
z
```

Corrected Symbolic Tape:

```
abcd    ldf   qxr
        add   xyz
        jfi   02
              next
        aod   stu
        zjf   05
        hlt   00
        end
```

11. Timing: Limited by paper tape reader and paper tape punch.

12. Cautions to user:

    a. Tape to be corrected must begin with a carriage return.

    b. Tape containing the corrections must begin with a carriage return.

    c. Any character in the additive field of the correction specification line that is not of the set (null, space, digit, delete code) will cause an error stop.

    d. The additive field of the correction specification line may be terminated only by a carriage return.

    e. The correction specification line identification, i.e., the symbol and additive fields, is always relative to the original program listing.

    f. When the block delete code immediately follows a d or r code, the n lines <u>following</u> the line referenced by the d or r code will be deleted.

    g. When the block delete code immediately follows an i code, n lines <u>including</u> the line referenced by the i code will be deleted.

    h. There may be any number (within the capacity of the Edit Program) of consecutive insertions or replacements <u>following</u> a single insertion or replacement code.

    i. A replacement of line n with m new lines is equivalent to deleting line n and inserting the m new lines at n + 1.

    j. An insertion results in a line or lines inserted AHEAD of the line specified in the correction code.

    k. The order of corrections (correction line identification) must be according to the original list.

    l. The symbolic identification of a line of the original listing must be made either by the location symbol of the line itself or by relative reference to the last symbol preceding the line.

    m. The same line in the original listing may not be referenced by more than one correction code.

    n. To facilitate editing symbolic tapes to be assembled by FLAP, a plus or comma symbol in the location Field of the original tape will be ignored as a tag.

14. Equipment configuration: Minimum SWAP 160 Computer

H1

IDENTIFICATION

Title: Magnetic Tape to Flex Tape - Ident. A3.02

Programmer: J. A. Pederson, April 1961

PURPOSE

Convert 80 character BCD coded information on magnetic tape in format of OSAP card input. This tape is suitable for listing on a Flexowriter and as input to the paper tape version of OSAP.

USAGE

1. Operational Procedure

   a. Load program tape

      1. Turn on reader and insert S 028
      2. Master Clear
      3. Set Load-Clear switch to LOAD
      4. Run (P=0000)

   b. Load magnetic tape

      1. Place tape to be converted on tape unit
      2. Select tape unit 1
      3. Select CODED mode
      4. Set to Load Point

   c. Turn on punch
   d. Set P = 0000
   e. Run
   f. Routine stops when end-of-file is read

3. Space Required: $330_8 = 216_{10}$ locations

7. Error Stops: Parity errors cause the routine to re-read tape three times before stopping.

13. Cautions to User: Inspect first frame on Flex tape output for carriage return (45).

A3.
IDENTIFICATION

Title: OSAP Symbolic Flex Tape to Cards - Ident. A3.03 (SYMCRD)
Programmer: J. Emmet Murphy, January 1962

PURPOSE:

To convert OSAP Symbolic Flex tapes to Hollerith cards.

USAGE:

1. Operational Procedure

   a. Machine load bioctal program tape at zero.
   b. Clear console and set any of the desired suppression codes in the A-register:
      4000 - suppress card sequencing
      2000 - suppress reading and converting flex tape (sequence cards only)
      If no suppression codes are used, the program will convert the Flex tape to sequenced, Hollerith cards.
   c. Load Symbolic Flex Tape to be converted and run.
   d. After final halt, load the next tape to be processed and continue as follows:

      1) For continuous sequencing, run from final halt (loc. 170_8).

      2) To re-initialize, clear, set parameters in A-register and run.

7. Error Stops:

   414_8 - too many suppression codes: suppress everything.

   463_8 - card punch not ready.

10. Timing:

    Maximum Punch speed.

13. Configuration:

    Minimum 160 or 160-A computer, 1610 and 523 Card Punch.

A4
IDENTIFICATION

Title: Binary Loader for FPP-33 Input - Ident. A4.11A
Programmers: D. A. Nelson, I. Hecker, July 1961

PURPOSE

This routine loads binary object program tapes prepared by FLAP (A1.02). Programs loaded by this routine are stored into memory and are ready for execution by FPP-33 (B1.00).

The binary object program tapes prepared by FLAP are relocatable. This means that if the programmer follows certain rules, then the FLAP binary object program can be loaded and executed from 160 memory locations other than those specified in the symbolic input.

USAGE

1. Operating Procedure:

   This routine is in 7400-7743 of FPP-33.

   a. Turn on reader.
   b. Position FLAP binary object program tape in paper tape reader.
   c. Set (A) = b, where b is the relocation constant.
   d. Set P = 7400
   e. Run
   f. Normal Stop:

      P = 7541 Z = 7700

      At the normal stop (A) = S where S is the 1604 address specified in the M-term of the END pseudo-instruction in the assembled program. If no address is specified, then S is zero. Normally S will be the starting address of the program to be simulated.

      If S is in the A register, then running from normal loader stop (7541) will enter the simulator and begin execution of the program just loaded.

3. Space Required: 7400 - 7743 of FPP-33.

7. Error Stops:

	P	Z	Reason
a.	7404	7701	Bias quantity (b) is greater than 1000_8.
b.	7416	7702	Incorrect binary tape label, i.e. the first non-blank frame encountered does not contain punches in levels 1 thru 6.

	P	Z	Reasons
c.	7507	7703	Current load address out of range. NOTE: FLAP is capable of assembling programs that cannot be run under control of FPP-33.
d.	7522	7704	Checksum error, bit run to ignore checksum error and go to normal stop.
e.	7766	7713	Illegal decimal number. Run to get error print-out as in simulator.

10. Timing:

The tape will be loaded at full tape speed unless DEC pseudo-ops were included in the assembly. When a decimal number is encountered, the loader must convert it to FPP-33 floating point format before storing.

13. Equipment Configuration - Minimum 160 Computer.

METHOD

The loader examines two frames of the FLAP binary output tape at a time. The seventh level bits are used to signify whether the information is a new load address, FPP-33 instructions to be stored, checksum, or a DEC number carried along by FLAP.

When a load address is encountered, the previous load address is replaced by that one. When 1604 instructions are encountered as binary information, they are stored sequentially starting at the last load address specified. If a DEC number is encountered, FPP-33 conversion routines are used to prepare it into FPP-33 floating point format and the result stored. The checksum consists of the last two frames on the binary tape.

AD1.01

IDENTIFICATION

Title: Floating Resident Service Library - Ident AD1.01
Programmer: A. Perro, December 1961

PURPOSE

Load memory from punched paper tape or the on-line electric typewriter; or dump the contents of the memory on punched paper tape or the on-line typewriter. The accuracy of the punched paper tape may be verified. Selection of input or output option is made at the console.

The following subroutines are callable from the resident service library:

Bi-Octal Punch	Flex Tape Verify
Bi-Octal Verify	Type Load
Flex Tape Load	Type Dump
Flex Tape Punch	

USAGE

1. Operational Procedure

   a. Load the program tape for RESERV as follows:

      1) Turn on the reader and insert the program tape anywhere on the blank leader.

      2) Lift load - clear switch in LOAD position and set P = to any desired location, X.

      3) Press run switch.

      4) Computer stops

         $P = X + 572_8$
         $A = 4210$

   b. Type load - used to store data received from the on-line electric typewriter.

      1) Turn on electric typewriter.

      2) Put operation mode switch into CLEAR position.

      3) Press operation mode switch into COMPUTER position. The ready light should be on and the input request light off.

4) Check to ensure input disconnect switch is in center position.

5) Set P = initial address of RESERV (X).

6) Press run switch. The status indicator will immediately show an input (IN) condition. The input request light and the ready light on the typewriter cabinet will also be on.

7) Type in instructions and data using the format described in the Flex Tape Load program.

8) Discrepancy - computer stops

   $P = X + 363_8$
   A = address of discrepancy
   Z = 0000

   To continue the program press the run switch after returning it to the center position.

9) End of file code - a period typed after the last carriage return causes the program to stop.

   $P = X + 304_8$
   Z = 7700

10) To repeat the program press the run switch after returning it to the center position.

c. Type Dump - produces listings of programs directly on the on-line electric typewriter.

1) Turn on typewriter.

2) Put operation mode switch in the CLEAR position.

3) Put operation mode switch in the COMPUTER position.

4) Check to ensure input disconnect switch is in the center position.

5) Set P = initial address + 3 of RESERVE (X + 3).

6) Press run switch.

7) After first stop set bank selection in right most 3 bits of A. Run.

8) After second stop set A = initial address to be dumped. Run.

9) After third stop set A = last address to be listed. Run.

10) Type dump will produce the specified listing allowing up to 48 individual listings on each page, with a double space preceding each listing group whose initial address is divisible by eight. Each listing contains a tab followed by (4 digits) which is the information. Address locations appear only at the beginning of each 10_8 group of listings. Computer stops to allow the operator to change paper.

11) End of Page

$P = X + 546_8$
A = 0000
Z = 7700

Continue by pressing the run switch after returning it to the center position.

12) Program stop - typewriter shows listing is completed by an end of file code (carriage return followed by a period).

$P = X + 567_8$
A = 7677
Z = 7700

d. Floating Flex Load - load previously prepared Flex coded paper tape in specified memory locations.

1) Turn on reader and insert tape to be loaded somewhere on the blank leader.

2) Set P = initial address + 6 of RESERV (X + 6).

3) Press run switch - program will store and check the information following the first carriage return character on the tape.

4) Computer stops on finding a discrepancy

$P = X + 363_8$
A = address at which discrepancy occurred
Z = 0000

5) Program stop

$P = X + 304_8$

Characters 0-7, carriage return, tab, and period are recognized by the program. All other non-printing characters are ignored by the program. A carriage return defines the beginning of a line.

Tape may be prepared in a nine digit format giving the bank designator followed by the four digits of the address followed by the four digits of the instruction or constant. A space or tab may or may not be inserted between the address and the data in this format. An eight digit format may be used but a TAB MUST SEPARATE THE ADDRESS FROM THE DATA and the program will store the information in the INDIRECT BANK which the operator manually sets. The program always takes the last four characters and stores them at the address specified by the first four characters.

Tape may also be prepared in a four digit format in which a carriage return followed by a tab followed by four characters of data make up a line. The program will take the four characters and store them at the next consecutive memory location following that at which information was last stored. If an insufficient number of digits in either format is present on a line, the data on that line will be ignored. A period appearing after a carriage return or a tab is interpreted as an end of file code and will cause the program to stop.

The following is an example of format supplied to Flex Tape Load:

Address	Instruction	Comments on Program
1463	3051	
	4163	
	5410	
1500	6550	
	4310	
4306	0112	
5060		
	0277	
	1793	
		• (period causes program to stop)

The Flex Tape Load program would cause the following information to be stored in memory:

Location	Contents
1463	3051
1464	4163
1465	5410
1500	6550
1501	4310
4306	0112
5060	0277

The last entry is missing because the program ignored the 9 and rejected the data because only three recognizable characters were left on the line.

e. Flex Tape Punch - produce paper tape listings of a program suitable for off-line printing on a Flexowriter or re-entry into the computer via the Flex Load option.

1) Turn on punch.

2) Set P = initial address + 14 of RESERV (X + 14)

3) Press run switch

4) After first stop (X + 413) set bank selection in A. Run.

5) After second stop (X + 421) set A = initial address to be listed. Run.

6) After third stop (X + 424) set A = last address to be listed. Run.

7) Program stop

P = X + 567

Flex tape punch will produce the specified listings allowing up to 48 listings on each page, with a double space preceding each listing with an address divisible by eight. Each listing contains a tab followed by (4 digits) which is the information. Address locations appear only at the beginning of each 10_8 group of listings.

f. Flex Tape Verify - verify the accuracy of a tape prepared by Flex Tape Punch.

1) Turn on reader

2) Insert Flexowriter tape into reader someplace on the blank leader.

3) Set P = initial address + 11 of RESERV (X + 11)

4) Press the run switch. On encountering a carriage return the program will begin assembling information according to the format rules given under Flex Tape Load. It will check this information against the current contents of the specified area memory. To verify a previously prepared tape proper indirect bank selection may be made.

5) Discrepancy - computer stops

$P = X + 363_8$
A = address of discrepancy
Z = 0000

Press run switch after returning it to center position. Program will continue to check the rest of the tape.

6) Program stop

$P = X + 304_8$
Z = 7700

g. Floating Bi-Octal Punch - punch out portions of the 160-A memory bi-octal format.

1) Start punch motor

2) Master Clear

3) Manually set the last octal digit of A to the memory bank from which you wish to punch data

4) Set P = initial address + 17 of RESERV (X + 17)

5) Press Run switch

6) Computer stops

P = X + 44
Z = 7700

7) Set A = initial address of the region to be punched

8) Press run switch after returning it to the center position

9) Computer stops

P = X + 47
Z = 7700

10) Set A = last address to be punched

11) Press run switch after returning it to the center position

12) Program stop

A = X + 36
Z = 7700

h. Floating Bi-Octal Verify - verify accuracy of a tape prepared by Bi-Octal Punch.

1) Set P = initial address + 21 of RESERV (X + 21)

2) Set A = first data word location on the tape; set indirect bank to the bank where data is to be verified

3) Turn on reader and insert tape

4) Press run switch

5) Discrepancy stop

P = X + 143
A = address of discrepancy
Z = 0000

6) Format error on the tape

P = X + 141
Z = 0

The tape must be removed because the computer will not accept the remainder.

7) Program stop

P = X + 104
Z = 7700

2. Space Required: $572_8 = 378_{10}$ locations

3. Timing

Output, approximately 110 frames per second
Input, approximately 350 frames per second

4. Cautions to User: Off line prepared Flexowriter tapes must start with a carriage return, code 45; input from typewriter must also start with a carriage return.

5. Equipment Configuration: Standard 160-A Computer.

D1.

IDENTIFICATION

Title: Magnetic Tape Bootstrap 160-A - Ident. AD1.02
Programmer: David W. McIlhenny, May 1962, Fleet Numerical Weather Facility

PURPOSE

The bootstrap program is used to keep itself and programs or data on magnetic tape (163-2 units) in order to eliminate paper tape handling. The bootstrap may be loaded and blocks written or read by simple console entry of pertinent data.

USAGE

1. Operational Procedure

a. To load bootstrap from magnetic tape:

(1) Magnetic tape with bootstrap rewound on unit 4

(2) Set tapes to binary

(3) Enter the following sequence beginning at location 0000:

7500
2134
7203
7400
7700
7100

(4) Clear, start at 0000

b. To load bootstrap from paper tape:

(1) Clear, load paper tape at location 7100 using binary loader.

c. To <u>write</u> a record on tape:

(1) Clear, P = 7100 set A = bank number, start
(2) P = 7110 set A = first address of block, start
(3) P = 7113 set A = last address + 1 of block, start
(4) P = 7117 set A = record number, start

d. To <u>read</u> a record from tape:

(1) Clear, P = 7220 set A = record number, start

3. Space Requirements

277 words octal
191 words decimal
program is assembled in bank 0 7100-7377

7. Stops

a. 7332 - Final stop both read and write

b. 7307 - Read end of file

On Read: This block is not on the tape.
On Write: There are not enough records to make this the one desired.

c. If tape rocks continually, the program is having difficulty in reading or writing the record desired. Put stop switch 4 up to stop rocking. The program will stop at P = 7271 with the type of error in the A register:

A = 0004 parity
A = 0020 read end of file; record is not there
A = any other; check sum is bad on read

9. Tape Mounting

The bootstrap uses tape unit 4 in binary.

12. Cautions to User

a. The bootstrap cannot be used to load information into locations 7100-7377 of bank 0. It may be reassembled to do so, but must be in bank 0.

b. Do not try to insert a block in the middle of a series. For example do not rewrite block 3 of a tape containing 5 blocks; the ones subsequent to 3 would be destroyed.

c. Do not step through program.

14. Equipment Configuration: Minimum SWAP 160-A and 163 Tape

METHOD

1. A maximum of safeguards, such as extra waits, are incorporated for dependability. Records are checked for both lateral parity and longitudinal checksum.

2. The bootstrap writes itself on the tape as a "flag" record when writing block 0. It searches for itself in order to synchronize when reading. Blocks are written with a 5 word identification record preceding each one. An end of file is written after a block when the write operation is performed.

D2.
IDENTIFICATION

Title: Simultaneous Card-to-Tape, Tape-to-Print Program (SIMO)
Ident. AD2.02
Programmer: Larry Brown, June 1962, Control Data Corporation

PURPOSE

Provides a peripheral processing system which fully utilized the capabilities of the 160-A Computer by running a card-to-tape operation simultaneously with a tape-to-printer operation. About 90% of maximum speed is obtained.

USAGE

1. Operational Procedures

Load the bi-octal tape into bank 0 starting at 0000. Master Clear, start at 0000 with OWXY in the A-register as in 2 below.

2. Parameters

OWXY as follows:
O = 0
W = 1 Card-to-tape
= 0 no card-to-tape

X = 1 Tape-to-print
= 1 no tape-to-print

Y = 0 use carriage control for tape-to-print
= 1 single space tape-to-print
= 2 double space tape-to-print
= 3 triple space tape-to-print

The Y parameter may be entered even though tape-to-print is not selected (i.e., X = 0). This allows subsequent starting of the tape-to-print operation by means of jump switch control (See 10.c.2.).

At each program stop A = 0000. A new OXYZ parameter may be inserted in A at that time, or A may be left 0000 to continue using the previous parameter. See Appendix 4 for full discussion of program stops.

3. Space Required

$4125_8 = 2133_{10}$ locations in bank 0
$1464_8 = 820_{10}$ locations in bank 1

5. Bank Allocation

Upon entry to the routines the direct, indirect, and buffer bank settings have been set to 1, and the relative bank is 0. Do not change these bank settings.

7. Program stops

Ten program stops are provided to indicate various error conditions, jump switch settings, and end-of-operation conditions. See Appendix 4 for discussion of stops.

9. Tape Mountings

a. Output tape, for card-to-tape operations, selection 1.
b. Input tape, for tape-to-print operations, selection 2.

10. Formats

a. Input, card-to-tape

1. BCD cards. Standard 80-column card with punching as per standard Hollerith set.

2. Binary cards. STandard column binary card, 7-9 punch in column 1.

3. End-of-File Card

Column 1 and 2 7-8 punches
Column 3 and 4 12-1-4-7 punches
Columns 5 through 5 blank

The end-of-file card terminates the card-to-tape operation, after causing an end-of-file mark to be written on tape 1. (During simultaneous operations the tape-to-print portion will continue.) See Appendix 1 to change the end-of-file card configurations. Cards are read from primary read station only.

b. Output, card-to-tape

1. BCD card images

80 character BCD image in even parity plus 8 look-ahead characters as follows:

20 20 20 20 20 20 20 20 if the next image on tape is a BCD image in even parity.

11 11 07 07 20 20 20 20 if the next image on tape is binary image in odd parity.

2. Binary card images

160_{10} character binary image in odd parity plus 8 100k-ahead characters as follows

00 04 00 00 00 01 00 00 if the next image on tape is a BCD image in even parity

00 05 00 01 00 05 00 04 if the next image on tape is a binary image in odd parity

See Appendix 2 for discussion of changing the character size of the images, and of changing or eliminating the look-ahead characters.

3. Input format, tape-to-print

Maximum 120_{10} character records in even parity. An end-of-file mark or end of tape reflective marker will stop the tape-to-print operation (during simultaneous operations the card-to-tape portion will continue.) If the carriage control feature of the program is used, the first character of the record is the control character.

4. Output format, tape-to-print

Maximum 120_{10} character line(same length as input record) printed. The first tape character in print position 1, the second in print position 2, etc. The first character is printed as blank if carriage control mode is selected.

5. Carriage control character.

The program recognized the following carriage control characters.

01_8	Skip to Format Channel 8 (page eject)
02_8	Skip to Format Channel 7
12_8	Double space
20_8	Single space
40_8	Triple space

All spacing and channel skipping is interpreted as spacing before printing.

Provision has been made to allow Monitor channel skipping and other interpretations of carriage control characters. See Appendix 3 for full discussion.

c. Selective Stop and Selective Jump Switches

1. Selective Stop Switches: Not used.
2. Selective Jump Switches:

1 = in the Up position program will start simultaneous operations if a card-to-tape only or a tape-to-print only operation is in progress. For this option to be available, Jump switch #1 must be at least momentarily in the neutral position.
#2 - Not used
#4 - in the UP position,program will stop. See Appendix 4.

11. Timing

a. Card-to-tape operations: 650 cards per minute.
b. Tape-to-print operations: 1000 lines per minute, maximum.
c. Simultaneous operations proceed with an overall speed of approximately 90-95% of maximum.

13. Cautions to User

Do not step through this routine. Do not take computer out of RUN.

To stop the program use jump switch #4 as in 10.c.2.

14. Equipment Configuration

160-A	-	Minimum SWAP Configuration
163-2	-	on normal channel
1612	-	on normal channel
1610	-	(with 088 card reader) - buffer channel

Tapes are written and read in character (6-bit) mode only.

A Note on the Programming Appendices

Provision has been made for this program to be easily adapted to meet the requirements of any installation. Changes may be made to the routines for 1) the end-of-file card, (2) the look-ahead characters and tape record length, and (3) tape-to-print carriage control interpretations. Each of these has been placed near the end of the program so that they may be found easily by the programmer.

Upon entry to the routines the direct, indirect, and buffer bank settings have been set to 1, and the relative bank is 0. Do not change these bank settings. Except where noted, contents of the A-register upon entry or leaving these routines need not be guaranteed.

Temporary storage for the routines may be made available by reserving locations in the relative bank 0, and referencing them with relative forward and relative backward mode instructions. Direct storage locations 00 and 77 are also available, as these are not used by the main program. In addition, at least two other direct storage locations may be changed by the programmer, as noted in the appendices. Unless otherwise allowed, changing of other locations is not permitted.

APPENDIX I

Programming for the End-of-File Card

The End-of-File card must appear as an illegal BCD card, that is, not both rows 7 and 9 in column 1 may be punched at at least one non-Hollerith punch combination must appear on some card column.

When the illegal punch combination is recognized, the program transfers to location CDCK9 to determine if this is an End-of-File card or an illegal BCD card. The check is to be made on the actual card image in memory. Direct storage locations TLOC contains the address of the first 12-bit word (9 row left) of the card image. The card image is located in 84 consecutive addresses and is to be examined to determine if it is that of an End-of-File card.

If the card image is that of an End-of-File card, return to the main program at location EOFCRD; otherwise return to location ERRCRD.

Direct storage locations available for changing: CERR and TEMP2.

APPENDIX 2

Programming for look-ahead characters and tape record length.

The length of the tape record written by the card-to-tape portion of the program is composed of two parts: (1) the number of characters in the tape-image - 80 for BCD records and 160_{10} for binary records, plus (2) the number of look-ahead characters. The tape record character length is kept in direct storage locations BCDLNG for BCD records and BINLNG for binary records. These direct storage locations are set during normal start at 0000 procedures, at locations (0)0010 and (0)0013. The record length may be changed by the programmer, but records longer than 176_{10} characters are forbidden.

The main program determines the card type (BCD or binary) of the current and next card image and following the conversion to a tape image transfers to one of the four following routines for insertion of look-ahead characters:

Location

BCDBCD - if a BCD card image will follow current BCD card image
BCDBIN - if a binary card image will follow current BCD card image
BINBIN - if a binary card image will follow current binary card image
BINBCD - if a BCD card image will follow current binary card image

The look-ahead characters are to be stored in consecutive locations, 1 character to a computer word, placed in the low-order 6 bits of the word. (Tapes are written in the character mode only.) Direct storage location STORE, upon entry to the above routines, contains the location to store the first character.

It is not necessary to have any look-ahead characters; the tape record length may be set to 80 characters for BCD cards or 160_{10} characters for binary cards. After setting the look-ahead characters, the routine must return to program location CDCK1.

Direct storage locations available for changing: STORE and TEMP2.

APPENDIX 3

Programming for control of the 1612 printer carriage control tape.

If carriage control of the tape-to-print portion of the program is selected (parameter Y = 0 as in C.2 above), the program transfers to location TCKENT for each tape record read with the control code in both the A-register and direct storage location TCKSVE.

The TCKENT routine is to interpret the control character and return to the main program at location TCCHK5 with a parameter in the A-register as follows:

A = 0000 - single space
A = 4000 - skip to format Channel 8 (new page)
A = 2000 - skip to Format Channel 7
A = 1000 - skip one line (double spacing)
A = 0400 - skip two lines (triple spacing)
A = 00P0 - skip to Monitor Channel P

All codes are interpreted as spacing before printing. Normal single spacing always follows every line printed. Skipping to Monitor Channels 1-6 is for the furrent tape record only; the selecti on is removed for subsequent records.

If carriage control mode is not selected (i.e., if parameter Y=1, 2 or 3 in C.2 above) the TCKENT routine is not entered. Instead the program maintains an internal count of the number of lines per page and page ejects when necessary. The number of lines used per page is stored in direct storage location LINES and is set during normal start at 0000 procedures at location (0) 0005. This parameter may be changed if required.

APPENDIX 3 (Cont.)

Stop Location	If A=0000 upon restart will continue with	Reason for start
	FROM CARD-TO-TAPE ONLY OPERATIONS	
0063	Card-to-tape	Tape write EOT or bad spot on tape
0066	Card-to-tape	E-O-F card or illegal BCD card read
0071 *	Card-to-tape	Jump switch 4 set
	FROM TAPE-TO-PRINT ONLY OPERATIONS	
0074	Tape-to-print	Read tape EOF or EOT
0077	Tape-to-print	Jump switch 4 set
	MISCELLANEOUS	
0102	Executes a start at 0000 operation	Illegal parameter (W=X=0)

* For the two stops at 0060 and 0071 a card has been read and its image is in memory. If A is non-zero upon restart and if parameter W=0, the card image will not be processed. A start at 0000 sequence will always cause loss of any information in core memory.

APPENDIX 4

Program Stops

Any tape-to-print operation started by the program has been completed at the time of all program stops. All non-end-of-file tape records read have been printed and the read tape is in position to continue reading with the next record. If an end-of-tape reflective marker has been encountered, the tape is in a rewind unload status.

A card has been read and its image is in core, but has not been processed at the time of the jump switch 4 stops (0060 and 0071). At all other stops no useable card image is in memory.

Following the processing of an end-of-file card, the write tape is positioned immediately after the end-of-file mark written. After the error conditions have been noted, the loss of information is inevitable and the position of the write tape or of cards in the card reader cannot be predicted. If an end-of-tape reflective marker has been encountered, the tape is in a rewind unload status, but at least one record has been written beyond the end-of-tape marker.

At each stop the A-register is 0000. If the A-register is 0000 upon restart the program will continue as per the chart below. If the A-register is non-zero, the value will be interpreted as a new OWXY parameter (see 10.c.2. above).

Stop Locations

	If A=0000 upon restart will continue with	Reason for stop
	FROM SIMULTANEOUS OPERATIONS	
0047	Simultaneous	Tape-to-print over and either tape write EOT or bad spot on tape.
0052	Tape-to-print	Tape write EOT or bad spot on tape
0055	Simultaneous	Tape-to-print over and either E-O-F card or illegal BCD card read.
0060 *	Simultaneous	Jump switch 4 set

D2.
IDENTIFICATION

Title: Peripheral Integrated Utility System (PIUS) - Ident. AD2.03
Programmer: E. J. Rosenberg, July 1962, System Development Corporation

PURPOSE

To provide a program to correlate and control a library of 160A routines in an executive utility system.

USAGE

1. Operating Procedures

   A. General (applies to any of the following control methods)
      1. Load P.T. Bootstrap at 7750 in Bank 0.
      2. Mount PIUS Master Tape on logical tape drive 2.

   B. Typewriter Control
      1. Set selective jump 1.
      2. Clear and run.
      3. Type in 4 character name of desired routine. If less than 4, terminate with a period.

   C. Card Control
      1. Place control card in card reader
      2. Make card reader ready
      3. Clear and run.

   D. Program Control
      1. Either the subroutine "CALL" (see appendix 1) or one of similar nature must be assembled with the controlling program.
      2. The following locations in bank 1 must be respected:

         70 = Any parameters necessary in "A" upon entrance to requested routine.
         71 = 0 - to preclude card read selection
         72 = BCD code for 1st 2 characters of requested routine (assembly mode)
         73 = BCD code for 2nd 2 characters of requested routine (where name is of less than 4 characters substitute 0's in lieu of BCD 20 code)
         74 = 1 - flag to PIUS to indicate program control
         75 = Relocation address of requested program (where a applicable).

3. Program utilizing this calling sequence must consider bank selection (as listed in the directory) of desired routine when jumping to it. The relative bank of the called routine must be the same as that of the controlling program.

4. In the case of OSAS store the I/O configuration selections in 72 and 73, i.e., they become the routine name.

MESSAGES

A. Typewriter

1. NOT A CONTROL CARD. No asterisks in column 1. Make corrections and run.
2. PROGRAM REFERENCE ERROR. The program requested is not in directory. Try again.
3. READY THEI/O EQUIPMENT AND RUN. I/O routine has been selected. Stop and wait for magnetic tape, etc. Run to continue.
4. ENTER PARAMETERS. While under typewriter control, you must key into "A" desired parameters and run.
5. RELOCATE AT. Type in relocation address and run. If none, run.
6. TYPE IN YOUR I/O SELECTION AND RUN. OSAS has been requested. Type in your I/O configuration and run.

B. Printer

1. Control card image.

ERROR STOPS

3511 - A card other than a control card has been read. Clear the card reader, correct the control card, return to reader, and run.

NORMAL STOPS

3706 - Stop to allow I/O selection to be made. Run to continue.

3745 - Stop to allow input of Parameters. Run to continue.

TAPE ASSIGNMENTS

The PIUS MASTER will be assigned to logical tape drive #2. Consists of 1 file with X + 2 records. (X = # of separate routines and programs included.) The first record is PIUS Control, the second record is the directory.

CONTROL CARD

Format:	Columns	Contents
	1	Asterisks
	3-6	Routine name as designated in directory
	9-12	Entrance parameters
	25-28	Relocation address

JUMP AND STOP KEY SELECTIONS

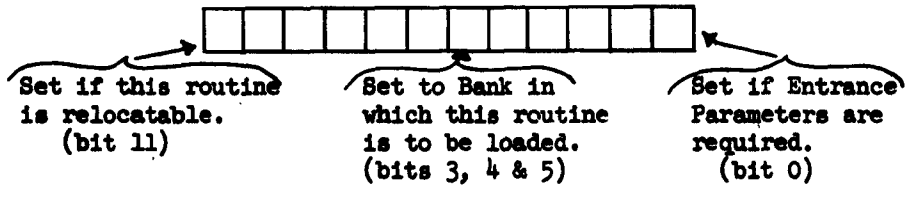
SLJ 1 - Typewriter control
SLJ 4 - Suppresses control card printing on 1612

METHOD

The system consists of 2 control records on a magnetic tape and 1 paper tape bootstrap of 52_8 locations. The bootstrap is loaded in high core of Bank 1 and operated.

This bootstrap loads the PIUS control program into Bank 1 and operates it. PIUS calls in the 2nd record, the directory, which contains seven words for each record on the master tape in the following format:

Word	
1	First 2 BCD characters of the program name.
2	Second 2 BCD characters of the program name.
3	Loading address
4	Last word address + 1
5	Complement of the record number
6	Beginning address
7	Routine Qualifications Viz:



It searches this directory for the routine name requested and reads that routine into the computer as prescribed by the remaining five words of the directory.

RESTRICTIONS

A. Requires a minimum 160-A with 163 tape cabinet and 161 typewriter.

B. Any routines written to be subject to PIUS control must not use the following locations:

0 - 1	Bank 0
70 - 77	Bank 1
7750 - 7776	Bank 0

C. Must not be programmed so as to be loaded above location 5700 of Bank 1.

STORAGE REQUIREMENTS

A. Program 427 locations of bank 1 (PIUS).

B. Storage 200 locations of bank 1 (Directory).

C. Storage 27 locations of bank 1 (Direct Cells-PIUS).

REFERENCES

A. PIUS Master Magnetic Tape-AF/CPL #

B. Symbolic Deck AF/CPL # Working ID

C. Assembly Listing AF/CPL #

APPENDIX I

```
          MAIN PROGRAM
                 :
          JPR  CALL          RETURN JUMP TO CALLING SEQUENCE SUBROUTINE
          LDS                SPECIFIC WILL CONTAIN THE BEGINNING ADDRESS
          STF  2                OF THE REQUESTED ROUTINE
          JPR  XXXX          RETURN JUMP TO ROUTINE JUST LOADED INTO STGE.
                 :
          CONTINUE
                 :
          JFI  1             RETURN TO MAIN PROGRAM
CALL           0             RETURN JUMP ADDRESS
          CTA                CONTROLS TO "A"
          LPN  7             BANK CONTROL INTERROGATION
          ADN  10
          STF  REL
          CTA
          LS6
          LPN  7
*         STF  BUFF
          CTA
          LS6
          LPN  7
          ADN  40
          STF  DIR
          CTA
          LPN  70
          ADN  20
          STF  IND
          LDD  METERS        IF OBJECT PROGRAM REQUIRES PARAMETERS STORE
          STD  70            THEM IN LOCATION 70 OF BANK 1.
          LDN  0
          STD  71
          LDC  4724          BCD CODE FOR CHARACTERS PU
          STD  72
          LDC  4400          BCD CODE FOR CHARACTER M ONLY
          STD  73            (0 SUBSTITUTED FOR BLANKS)
          LDN  1
          STD  74            SET FLAG FOR PIUS TO RECOGNIZE PROGRAM CONT.
          LDC  3000          STORE THE RELOCATION ADDRESS (WHERE APPL.)
          STD  75
BOOT      EQU  7751          THE 2ND LOCATION OF THE BOOTSTRAP
          PTA
          ADN  3
          ACJO               SET ALL BANKS TO 0 TO OPERATE PIUS BOOTSTRAP
```

*If bank selections are previously known this area may be dispensed with.

APPENDIX I (continued)

```
        JPR  BOOT
IND          0
DIR          0
BUFF         0
        LDC  CALL    1      RESET INDIRECT BANK
             0              THE SUBROUTINE HAS STORED AN SRJ INSTRUC-
                              TION HERE
```

E2.

IDENTIFICATION

Title: 167 Card Read Routine - Ident AE2.01
Programmer: A. R. Perro, October 1962, Control Data Corporation

PURPOSE

Reads a Hollerith or binary card with the 167 Card Reader and stores the information either packed (12 bits per word) or unpacked (6 bits per word) in the lower half of each word. The routine allows 1-80 columns to be read from a card.

USAGE

1. Calling Sequence:

```
L-1  LDN            PARAM
L    JPL            CARDIN +1
L+2  Starting location of input
L+3  Terminating location of input
L+4  Return
```

2. Parameters:

```
PARAM = 000X        X = 0 BCD          X = 1 Binary
      = 00Y0        Y = 0 Packed       Y = 1 Unpacked
```

3. Space Requirements: $243_8 = 227_{10}$ locations.

11. Timing: 1.5 milliseconds times number of columns per card plus 70 milliseconds latch up.

12. Caution to User:

No provision for bank settings is made in the routine.

14. Equipment configuration: SWAP 160-A computer, and 167 Card Reader.

F1.
IDENTIFICATION

Title: Punched Paper Tape Duplicator - Ident. F1.00
Programmer: J. Pederson, December 1960

PURPOSE

This program produces multiple copies of a given seven level punched paper tape.

USAGE

1. Operation Procedure

   a. Enter Tape

      1. Turn on reader, insert F1.00 tape and load starting P = 0000
      2. Place tape to be duplicated on the reader and enter with P = 0000. Tape will be read in until succeeding blank frames indicate the end of the tape.
      3. Program Stop
         P = 0214
         A = 0000
         Z = 7701
      4. If more information remains to be duplicated press switch after returning it to center position.
      5. Error Stop - More than 6395 frames have been entered.
         P = 0206
         A = 7777

   b. Verify

      1. Place tape on reader and run with P = 0001
      2. Program Stop
         P = 0034
         Z = 7707
      3. Error Stop
         P = 0361
         A = 0000
         Z = 7705

   c. Punch

      1. Turn on Punch
      2. Set P = 0002 A = number of copies desired (in octal)
      3. Press Run switch

4. Program Stop
   P = 0320
   Z = 7707
   The program will punch an 18 inch leader between copies.

d. To Verify multiple copies

1. Place tape in the reader
2. Set P = 0001; A = number of copies to be verified
3. Error Stop
   P = 0361
   Z = 7705
   Tear out bad copy. Master Clear. Place tape on the reader and set P = 0030. Press Run switch. Alternative method: Set P = 0001, A = remainder of the tape to be verified and press Run switch. Program Stop with
   P = 0034
   Z = 7707
   The program will copy up to 6395 frames of seven level tape and produce the number of copies specified by the A Register. It also produces an 18 inch leader between copies. Use F1.01 for tapes greater than 6395 frames.

3. Storage Requirements: Program 0000 - 0377
   Tape Image 00400 - 7775

13. Equipment Configuration - Minimum 160 Computer.

F1.

IDENTIFICATION

Title: Duplicate Paper Tape - Ident. AF1.00

Programmer: J. A. Pederson, August 1960

PURPOSE

This routine duplicates information on paper tape from the photoelectric reader to the high speed punch.

USAGE

1. Operational Procedure - The routine floats and may be loaded anywhere in memory.

   a. Load, set P = any arbitrary address.

```
7507  select reader
7210  read 2 frames
0072
7505  select punch
7305  punch 2 frames
0072
6506  go back
4102  reader code
4104  punch code
0070  start of I/O
```

   b. Master Clear

   c. Set P = starting address selected

   d. Turn on reader and punch

   e. Place tape to be duplicated in the reader

   f. Press Run switch

   g. To stop duplication return switch to center. The duplication process stops automatically after the tape in the reader has been read.

3. Space required - $12_8 = 10_{10}$ locations

13. Equipment Configuration - Minimum 160 computer.

F1.

IDENTIFICATION

Title: Punched Paper Tape Duplicator - Ident. F1.01
Programmer: R. M. Olson, February 1961

PURPOSE

This routine is designed to reproduce paper tapes of more than 6395 frames by placing the paper tape image on magnetic tape.

USAGE

1. Operational Procedure

   a. Load Program Tape S 025

      1) Place S 025 tape in the reader and set P = 0050.

      2) Set load switch and press run.

      3) Program stop

         P = 0406
         A = 2161
         Z = 0000

   b. Enter Tape to be Duplicated

      1) Place tape to be duplicated in the reader.

      2) Load a reel of magnetic tape onto tape unit. Check the following items:

         a) Magnetic tape unit 1 selected.

         b) BINARY parity selected.

         c) The tape is run forward past load-point, or is set to load point and the CLEAR button has been pushed.

         d) The magnetic tape unit has a green ready light on.

         The magnetic tape units need not be touched during the rest of the operations.

3) Set P = 0050

4) Press Run switch

5) Temporary program stop

P = 0167
A = 0000
Z = 7700

At this stop there are two options:

a) If more tape is to be duplicated; press the Run switch after returning it to the center position.

b) If at the end of the tape: Master Clear, set P = 0166, and press the Run switch. This will transfer the last partial block of data onto magnetic tape.

6) Final program stop.

P = 0204
A = 2161
A = 7700

7) Error Halt (parity error)

P = 0207
A = 2161
Z = 7750

c. Verify Master Tape and Copies

1) Place tape in reader.

2) Set P = 0052

3) Press the Run switch

4) Proper verify stop

P = 0320
A = 2161
Z = 7700

5) Improper verify stop

P = 0316
A = 2161
Z = 7700

6) Error Halt (parity error)

P = 0306
A = 2161
Z = 7752

d. Punch New Copies

1) Turn on punch

2) Set P = 0054

3) Press the Run switch

4) Program stop

P = 0400
A = 2161
Z = 7700

5) Error Halt (parity error)

P = 0376
A = 2161
Z = 7754

3. Space Required - $324_8 = 212_{10}$ locations.

4. Temporary Storage Requirements - locations 0070 through 0076.

12. Caution to user - Make certain that the magnetic tape unit is set to BINARY mode and the Tape Unit 1 is selected.

13. Equipment configurations - Standard 160 computer.

F1

IDENTIFICATION

Title: Tape Leader Preparation - Ident. F1.02

Programmer: J. A. Pederson, August 1960

PURPOSE

Used to prepare tape leaders which identify program tapes. The characters 0 to 9, as S,T,P,R ,-,., can be punched on a tape leader.

USAGE

1. Operational Procedure

   a. Load program starting at location zero.

   b. Master Clear, turn on the punch and start. The program will punch two frames of leader.

   c. Place in A the desired character to be punched from the following table.

   d. Push Run switch. The program will punch the character and stop.

   e. After preparing the leader on the punch, run out about two inches and then the desired program may be copied.

      If the program is internal the resident service routine may be used to punch out the information from core storage. By clearing and starting at location 0050, the information on the photoelectric paper tape reader may be repunched on the high speed punch.

Character Code in A Register:

0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	10
9	11
	12
S	13
T	14

Character Code in A Register (Cont.)

P	15
-	16
R	17

3. Space required $311_8 = 201_{10}$ locations

4. Temporary Storage Requirements - 7 locations

10. Timing - Output approximately 60 frames per second

14. Equipment Configuration - Minimum 160 computer

F1.
IDENTIFICATION

Title: Tape Copy/Compare - Ident. F1.05
Programmer: John Steffani, August 1962, Control Data Corporation

PURPOSE

This routine utilizes the Control Data 160 or 160-A computer to copy or compare magnetic tape files on 163 or 164 tape units. The operator has the following options:

1. Copy or compare operation.
2. BCD or binary mode.
3. Rewind load at start.
4. Rewind unload at finish.
5. Up to $(77)_8$ files can be copied or compared per run.

NOTE: For operations involving more than $(77)_8$ files, the following procedure may be followed:

1. Enter control word in Direct Storage location 0067, (lower 6-bits zero).
2. Enter number of files to check indirect storage location 0064.
3. Set P to 0103.
4. Press RUN.

USAGE

1. Operational Procedure

a. Clear core, set all banks to zero.
b. Set P-register to 0100.
c. Load bi-octal program paper tape.
d. Mount magnetic source tape, remove file protection ring, set unit selection switch to #1.
e. Mount output or copy tape, set unit selection switch to #2.
f. Set both tapes to desired starting location (tapes must not be in a rewind unload state).
g. Enter control word into A-register.
h. Set P to 0100.
i. Set RUN-STEP switch to RUN.

2. Parameters

a. Control word bit structure.

Bits 0 thru 5	= set equal to number (octal) of files to be copied or compared.
Bits 6 and 7	= set to zero.
Bit 8	= set to 1 for rewind with lockout both tapes when number of files specified is completed.
	= set to 0 for no rewind.
Bit 9	= set to 1 for rewind load both tapes at start of operation.
	= set to 0 for no rewind at start.
Bit 10	= set to 1 for compare operation. Compare tape 2 against tape 1.
	= set to 0 for copy operation. Copy tape 1 onto tape 2.
Bit 11	= set to 1 for binary coded input/output
	= set to 0 for BCD coded input/output.

Example control word to compare 3 BCD files with rewind unload upon completion, rewind load at start:

1) Bit representation

11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	0	0	0	0	0	0	1	1

2) Octal representation

3403

3. Space Required: $(7776)_8$, $(4094)_{10}$ locations.

7. Error Stops

P-register	Explanation
0214	Record length too long for this routine. Discontinue operation.
0234	Parity error on Read Tape #1. To re-read, press RUN switch. To continue (with parity error present), a) Master clear. b) P-register to 0243. c) Press RUN switch.
0261	Compare operation, record length tape #2 too long for this routine. Discontinue operation.
0270	Compare operation: EOF TAPE 2, NO EOF TAPE 1. The above condition is the result of either of two causes: a) Tapes are not compatible, i.e., there are more records per file on tape 1 than tape 2;

for this case, operation should be discontinued.

b) The remainder of the file for the output tape is on a continuation tape. For this case a visual inspection of tape unit #2 will show the reel to be nearly empty. For this case, continue as follows:

1) Rewind unload tape unit #2.
2) Mount next output tape and place in a rewind load position.
3) Set P-register to 0220.
4) Press RUN.

0301 Compare Operation: Record length tape #1 not equal (≠) to record length tape #2. Tapes not compatible, discontinue operation.

0307 Compare Operation: Parity error on read tape #2. To re-read, press RUN switch. To continue (with parity error present):

a) Master clear.
b) Set P to 0317.
c) Press RUN.

0333 Compare Operation: compare error.

a) To backspace, re-read and compare again. Press RUN.
b) Word on tape #2 causing error displayed in A-register. Address of corresponding word on tape #1, located at 0040.
c) To by-pass compare error, set P to 0326.

1) Master clear.
2) Set P to 0326.
3) Press RUN.

0374 Copy operation: End of tape encountered on tape #2. Record has not been written on tape #2. EOF has been written.

a) Rewind unload output tape.
b) Mount next output tape and place in rewind-load position.
c) Press RUN.

0402 Parity error on attempt to write tape #2. To backspace and rewrite, press RUN.

0417 Copy operation: End of tape encountered on tape #2. Record has been written on tape #2, EOF has been written.

a) Mount next output tape and place in rewind load position.
b) Press RUN.

0443 Compare operation: End of file tape #1, no EOF tape #2. Discontinue operation.

0462 Operation completed.

10. Input and Output Formats

Input record length should not exceed the following:

a. Copy operation $(16,624)_8$ characters.

b. Compare operation $(7,310)_8$ characters.

14. Equipment Configuration

Minimum SWAP 160 or 160-A computer, 163-2 or 164-2 magnetic tape handler.

F1.
IDENTIFICATION

Title: Tape Compare (STACO) - Ident. AF1.06
Programmer: E. J. Rosenberg, August 1962, System Development Corporation

PURPOSE

To compare two BCD magnetic tapes to insure similarity.

USAGE

Operating Procedures:

1. Machine load bioctal paper tape at 0.
2. Select logical tape drives to 1 and 2.
3. Clear and run from 0.

Normal Stops

P = 213 A = 7701
One file has been compared without error. Clear and run to process additional files.

P = 220 A = 7700
All files on this tape have been compared, and tapes rewound to unload.

Error Stops

P = 141 End of tape on drive #1 has been sensed.
P = 221 There is a read parity on drive #1. After 5 attempts it has been unsuccessful.
P = 343 Halt on miscompared records.
P = 355 There is a read parity on TD #2. After 5 attempts it has been unsuccessful.

MESSAGES

Typewriter:	MISCOMPARE Indicates the latest read records were not the same.
1612 Printer:	The 1st of 2 lines printed is what was in the card image of the miscompared record of TD #1. The 2nd line is from TD #2.

JUMP & STOP KEYS
SLS 1 - will preclude rewind of tapes after 1 file has been compared.
SLJ 1 - will suppress printing of card images in cases where a 1612 is not available.

RESTRICTIONS

Tapes must be written in BCD card (6 bit hollerith) format.
Hardware
Minimum 160-A
161 Typewriter
1612 Printer (optional)

STORAGE REQUIREMENTS

423_8 Storage locations of bank 0.
6 Constants in direct bank.

F2.

IDENTIFICATION

Title: Flex Load and Flex Verify - Ident. AF2.00
Programmer: L. Kuller, September 1960

PURPOSE

Flex Load is designed to read 160 program tapes prepared on a Flexowriter or by the Flex Dump program and store them in memory at addresses specified by the tape.

Flex Verify is designed to check 160 program tapes prepared on a Flexowriter or by the Flex Dump program against the current contents of the 160 storage locations as specified by the tape. The program indicates any discrepancies that are found.

USAGE

1. Operational Procedure

   a. Turn on reader

   b. Insert tape someplace on blank leader

   c. Flex Load set P = 0400
      Flex Verify set P = 0401

   d. Press Run switch

   e. Discrepancy stop (Flex Verify)
      P = 0434
      A = address of the discrepancy
      Z = 0001

   f. Program Stop
      P = 0414
      Z = 7707

3. Space Required - $177_8 = 127_{10}$ locations

4. Temporary Storage Required - Locations 0072 through 0077

12. Caution to User - Location 7777 can not be loaded or verified by these programs unless it is referenced in the eight digit format. If 7777 is referenced in the four digit format the data will be stored at location 0000.

14. Equipment Configuration - Minimum 160 Computer

F3.

IDENTIFICATION

Title: Floating Bi-octal Punch and Floating Bi-octal Verify - Ident. AF3.00
Programmer: J. Oden, June 1961

PURPOSE

Floating Bi-octal Punch is used to punch out portions of the 160-A memory in bi-octal format. Floating Bi-octal Verify compares bi-octal date on punched paper tape with the current contents of the 160-A memory.

USAGE

1. Operational Procedure

Master clear and turn on the reader. Load the tape beginning at any location with the load-clear switch in LOAD position.

a. Floating Bi-octal Punch

1. Start punch motor

2. Master clear

3. Manually set the last octal digit of A to the memory bank from which you wish to punch data.

4. Set P = initial address of the floating bi-octal routine, "X"

5. Press Run switch

6. Computer stops

P = X + 12

Z = 7700

7. Set A = initial address of the region to be punched.

8. Press Run switch after returning it to the center position

9. Computer stops

P = X + 15

Z = 7700

10. Set A = last address to be punched.

11. Press Run switch after returning to the center position

12. Program stop

    A = X + 4

    Z = 7700

b. Floating Bi-octal Verify

1. Set P to the third location of the region containing the floating bi-octal routines: X + 2

2. Set A = first data word location on the tape

3. Turn on reader and insert tape

4. Press run switch

5. Discrepancy Stop

   P = X + 111

   A = address of discrepancy

   Z = 0000

6. Format error on the tape

   P = X + 107

   Z = 0000

   The tape must be removed because the computer will not accept the remainder.

7. Program stop

   P = X + 52

   Z = 7700

3. Space Required: $75_{10} = 113_8$ locations. Temporary storage requirements: None. This routine does not use addresses 0-77.

13. Equipment Configuration - Minimum 160-A computer

THEORY OF OPERATION

The instruction 002X = SIC, sets indirect storage bank control (i) to x. All subsequent indirect storage references address storage bank x. This value x is manually set in the last octal digit of A; then the program builds instruction SIC.

By using instructions LCN 77 and LPN 77 the 12 bit words are separated into upper and lower six bits. One is added to the word after the lower six bits have been masked and SHA 11 instruction given to get seventh level format. These are output immediately via OTA = 7677 to avert the problem of storage and establishing initial and terminal addresses in this floating routine.

In the Floating Bi-octal Verify, the seventh level format is checked, and also each word on the tape is compared with its counterpart in memory. Stops are provided in the program where format and word discrepancies occur.

F3.

IDENTIFICATION

Title: Memory Dump - Ident. F3.01
Programmer: G. M. Bronstein, August 1961

PURPOSE

Dump most of memory on magnetic tape.

USAGE

1. Operationa; Procedure

   a. Load program tape

      1. Turn on reader and insert S 030 tape

      2. Master Clear

      3. Set P = 7511

      4. Set Load-Clear switch to LOAD

      5. Run

      6. Stop - P = 7776

   b. Load magnetic tape

      1. Place magnetic tape on tape unit

      2. Select tape unit 1

      4. Select BINARY mode

      5. Set tape to Load Point

   c. Set P = 7511

   d. Run

3. Space Required - $265_8 = 181_{10}$ locations

13. Cautions to User: The routine dumps memory locations 0000-7757. There are no parity checks made in writing on tape.

14. Equipment Configuration: Standard 160 with 160 tape unit.

F3.

IDENTIFICATION

Title: Listable Symbolic Dump - Ident. F3.03
Programmer: Don T. Miller, July 1961

PURPOSE

To produce a listable symbolic dump of a specified area of core with 4 words per Flex line.

USAGE

1. Operational Procedure

   The program may be loaded anywhere greater than address 1. The P register should be set to the starting load address, the starting address of the area to be dumped put in the A register, and the Run switch put in the RUN position. The program will stop at the load address +7, and the final address of the dump placed in A. Put the Run switch in RUN and the program will punch the required area of core (in Flexowriter code), punch a stop code and stop at the load address $+47_8$.

3. Space Used: $436_8 = 281_{10}$ locations

9. Output Format: Each line will contain one octal address, the octal contents, the mnemonic instruction code and the last two octal digits of four words.

10. Timing: The program operates at punch speed.

12. Caution to User: The contents of 7777 will not be dumped. The program will operate if the final address is less than the start address.

13. Equipment Configuration: Minimum

F3.

IDENTIFICATION

Title: Listable Symbolic Dump - F3.04A
Programmer: Don T. Miller, July 1961

PURPOSE

To produce a listable symbolic dump of a specified area of core with 1 word per Flex line.

USAGE

1. Operational Procedure

   The program may be loaded anywhere greater than address 1. The P register should be set to the starting load address, the starting address of the area to be dumped put in the A register and the Run switch put in the RUN position. The program will stop at the load address +6, and the final address of the dump should be placed in A. Put the Run switch in RUN and the program will punch the required area of core (in Flexowriter code) punch a stop code and stop at the load address +41.

3. Space Used: $421_8 = 273_{10}$ locations

9. Output Format: Each line will contain: a carriage return, octal address, octal contents of that address, mnemonic instruction code for the first two digits and the last two octal digits, e.g., (0100 4201 STF 01).

10. Timing: The program operates at punch speed.

12. Caution to User: The contents of 7777 will not be dumped. The program <u>will</u> operate if the final address is less than the start address.

13. Equipment Configuration: Minimum

F3.

IDENTIFICATION

Title: 160-A Symbolic Printer Dump - AF3.07
Programmer: L. Brown, S. Palais, January 1962

PURPOSE

To provide a dynamic symbolic dump for the 160-A onto the 1612 printer.

USAGE

1. a. As a closed subroutine

```
L     JPR
L+1      DMPPRT
L+2   NORMAL
L+3      RETURN
L+4   000X (X = Bank to be dumped)
L+5   FWA of dump
L+6   LWA of dump
```

The calling sequence must be in the same bank as the subroutine. The A-register and bank settings will be restored by the program.

b. Independent Routine Procedure

(1) To locate the program at location (P), load the OSAP-A binary output with the OSAP-A relocatable binary loader, using (P) as the relocation increment. (P) must be $\leq 6770_8$. The program may be loaded into any bank.

(2) Master clear.

(3) Ready the 1612 printer.

(4) Put (P) in the P-register and 000X (where X is the bank to be dumped) in the A-register.

(5) RUN. Upon halting at (P+2), enter FWA in the A-register.

(6) RUN. Upon halting at (P+5), enter LWA in the A-register.

(7) RUN. The program will stop at (P+10).

(8) RUN with 000X in the A-register to restart.

3. Space Required

$1005_8 = 517_{10}$ as an independent routine.

$765_8 = 501_{10}$ as a closed subroutine.

9. Output Format

First Line

DUMP(n)XXXX TO (n)XXXX LOC(n)7777 = XXXX
BANKS XXXX AREG XXXX JUMP ADDRESS XXXX

where (n) is the number of the bank being dumped.

BANKS XXXX is the display of the bank settings, bdir, upon entry to the dump program.

AREG XXXX is the contents of the A-register upon entry to the dump subroutine.

JUMP ADDRESS XXXX is the address L in the calling sequence. (L is $P+10_8$ when program is used independently.)

Normal Lines:

The normal line output consists of an address followed by 10_8 words of the form: 3 digit op code and 4 digit octal word. The first address dumped will be the nearest address less than FWA and divisible by 10_8. The number of words per line may be made less than 10_8 by replacing location ($P+26_8$) by LCN N, where N is the word count.

The last line dumped will include, but not necessarily end with, the word in LWA.

Zero Lines:

1st - normal line format
2nd - .....
subsequent lines of zeros are suppressed.

Last Line:

END DUMP (n)XXXX TO (n)XXXX

NOTE: Location (n) 7777 is included <u>only</u> in the first line printed. If 777 is given as the FWA, 0000 will be the FWA used by the routine; if 7777 is given as the LWA, 7776 will be the LWA used.

No symbolic op codes are printed for 00 and 01 F-terms.

13. Equipment

160-A, 1612

F3.

IDENTIFICATION

Title: 160-A Symbolic Flexowriter Dump - Ident. AF3.08
Programmer: L. Brown, S. Palais, January 1962

PURPOSE

To provide a symbolic dump for the 160-A onto punched paper tape, to be listed on the flexowriter.

USAGE

1. a. As a closed subroutine

```
L     JPR
L+1      DMPFLX
L+2   NORMAL
L+3      RETURN
L+4   000X (X = Bank to be dumped)
L+5   FWA of dump
L+6   LWA of dump
```

The calling sequence must be in the same bank as the subroutine. The A-register and bank settings will be restored by the program.

b. Independent Routine Procedure

(1) To locate the program at location (P), load the OSAP-A binary output with the OSAP-A relocatable binary loader, using (P) as the relocation increment. (P) must be $\leq 6740_8$. The program may be loaded into any bank.

(2) Master Clear.

(3) Turn the paper tape punch on.

(4) Put (P) in the P-register and 000X (where X is the bank to be dumped) in the A-register.

(5) RUN. Upon halting at (P+2), enter FWA in the A-register.

(6) RUN. Upon halting at (P+5), enter LWA in the A-register.

(7) RUN. The program will stop at (P+10) after punching paper tape.

(8) RUN with 000X in the A-register to restart.

3. Space Required

$1036_8 = 542_{10}$ as an independent routine.

$1016_8 = 526_{10}$ as a closed subroutine.

9. Output Format

First line:

```
DUMP(n)XXXX TO (n)XXXX LOC(n)7777 = XXXX
  BANKS XXXX AREG XXXX JUMP ADDRESS XXXX
```

where (n) is the number of the bank being dumped.
BANKS XXXX is the display of the bank settings, bdir, upon entry to the dump program.

AREG XXXX is the contents of the A-register upon entry to the dump subroutine.

JUMP ADDRESS XXXX is the address L in the calling sequence. (L is $P+10_8$ when program is used independently.)

Normal Lines:

The normal line output consists of an address followed by 10_8 words of the form: 3 digit op code and 4 digit octal words. The first address dumped will be the nearest address less than FWA and divisible by 10_8. The number of words per line may be made less than 10_8 by replacing location ($P+26_8$) by LCN N, where N is the word count.

Zero Lines:

1st - normal line format
2nd - .....
subsequent lines of zeros are suppressed.

Last Line:

END DUMP (n)XXXX TO (n)XXXX

NOTE: Location (n) 7777 is included <u>only</u> in the first line punched. If 7777 is given as the FWA, 0000 will be the FWA used by the routine; if 7777 is given as the LWA, 7776 will be the LWA used.

13. Equipment

160-A, Paper Tape Punch

F3.

IDENTIFICATION

Title: Octal Dump - Ident AF3.09
Programmer: D. T. Miller and S. Palais, November 1961

PURPOSE

Provides an octal dump of one bank of 160-A memory from a given starting address to the end of that bank onto the 1612 printer. The program is relocatable.

USAGE

1. Operational Procedure

   a. The program may be loaded anywhere.

   b. Set the P-register to the starting load address and the A-register to the starting dump address.

   c. Set the indirect bank setting to the number of the bank to be dumped.

   d. RUN.

   e. Program Stop

   Load address + 40_8.

3. Space Required: $132_8 = 90_{10}$ in the relative bank.

4. Temporary Storage: Location 77 in the direct bank, which is not restored.

9. Output Format

   The line format consists of an address and 20 words. An asterisk indicates the omission of one or more lines of zeros. Duplicate lines will be printed, and the contents of the A-register will not be printed.

10. Timing

   For consecutive non-zero lines, the program operates at printer speed.

13. Equipment Configuration: Minimum 160-A Computer, 1612 Printer.

F3.
IDENTIFICATION

Title: 160-A Symbolic Typewriter Dump - Ident AF3.10
Programmers: L. Brown, S. Palais, T. Easley, June 1962, Control Data and Philco

PURPOSE

Provides a symbolic dump for the 160-A onto the 161 typewriter.

USAGE

1. a. As a closed subroutine

```
L     JPR
L+1      DMPTYP
L+2   NORMAL
L+3      RETURN
L+4   000X  (X=Bank to be dumped)
L+5   FWA of dump
L+6   LWA of dump
```

The calling sequence must be in the same bank as the subroutine. The A-register and bank settings will be restored by the program.

b. Independent Routine Procedure

(1) To locate the program at location (P), load the OSAS-A binary output with the OSAS-A relocatable binary loader, using (P) as the relocation increment. (P) must be $\leq 6544_8$. The program may be loaded into any bank.

(2) Master Clear.

(3) Turn the typewriter on.

(4) Put (P) in the P-register and 000X (where X is the bank to be dumped) in the A-register.

(5) RUN. Upon halting at (P+2), enter FWA in the A-register.

(6) RUN. Upon halting at (P+5), enter LWA in the A-register.

(7) RUN. The program will stop at (P+10) after typing out the last line.

(8) RUN with 000X in the A-register to restart.

3. Space Required

$1232_8 = 542_{10}$ as an independent routine.

$1212_8 = 526_{10}$ as a closed subroutine.

10. Output Format

First Line:

DUMP(n)XXXX TO (n)XXXX LOC (n)7777 = XXXX
BANKS XXXX AREG XXXX JUMP ADDRESS XXXX

where (n) is the number of the bank being dumped.
BANKS XXXX is the display of the bank settings, bdir, upon entry to the dump program.

AREG XXXX is the contents of the A-register upon entry to the dump subroutine.

JUMP ADDRESS XXXX is the address L in the calling sequence. (L is $P+10_8$ when program is used independently.)

Normal Lines:

The normal line output consists of an address followed by 4_8 words of the form: 3 digit op code and 4 digit octal word. The first address dumped will be the nearest address less than FWA and divisible by 10_8. The number of words per line may be made more or less than 4_8 by replacing location ($P+26_8$) by LCN N, where N is the word count.

The last line dumped will include, but not necessarily end with, the word in LWA.

Zero Lines:

1st - normal line format

2nd - .....

subsequent lines of zeros are supressed.

Last Line:

END DUMP (n)XXXX TO (n)XXXX

NOTE: Location (n) 7777 is included only in the first line punched. If 7777 is given as the FWA, 0000 will be the FWA used by the routine; if 7777 is given as the LWA, 7776 will be the LWA used.

14. Equipment Configuration: Minimum SWAP 160-A computer, 161 Typewriter.

F3.
IDENTIFICATION

Title: 160-A Core Dump to 1612 (1612 DUMP) - Ident AF3.11
Programmer: Richard Mihm, January 1962, Control Data Corporation
(Documented by E. J. Rosenberg, July 1962, System Development Corporation)

PURPOSE

To allow a listable dump of the machine codes and constants stored in 160-A and 169 onto the 1612 printer. Completely Relocatable.

USAGE

A. Operating Procedures:

1. Call DUMP from PIUS Master or load bioctal paper tape anywhere.
2. Run from load address.
3. Answer typewriter queries as to area and bank to be listed. Terminate area specifications with periods.
4. Run.

B. Stops:

1. Will stop at beginning of this program. Clear and run to return to PIUS control.

RESTRICTIONS

A. Hardware

1. Minimum 160-A
2. 161 typewriter
3. 1612 printer

STORAGE REQUIREMENTS

455_8 locations in bank 0.

F3.
IDENTIFICATION

Title: BiOctal Dump (BIDU) - Ident. AF3.12
Programmer: E. J. Rosenberg, August 1962, System Development Corporation

PURPOSE

Punches a given area of a selected bank in bioctal format for machine loading.

USAGE

1. Under PIUS Control:

   a. Call BIDU.
   b. At stop L*, set indirect bank equal to the bank in which your output is contained. Set "A" equal to the first word address of the output area and run.
   c. At stop L + 3, set your LWA + 1 in "A" and run.
   d. Normal stop is at location L.
   e. To return to PIUS control, clear and run.

2. Paper tape version:

   a. Machine load anywhere in bank 0. Run from loading address.
   b. Follow steps 1b, and following, above.

RESTRICTIONS

1. Must be 7 level paper tape.
2. Requires 37_8 storage cells in bank 0.
3. Minimum 160-A.

* L = Loading address.

F4.
IDENTIFICATION

Title: 1607 to 160 to 1612 Routine - Ident. F4.06
Programmer: Sanford Elkin, February 1962

PURPOSE

Accepts blocked or unblocked magnetic tapes, and lists them on the Control Data 1612 printer at 1000 times per minute. The listing may be done either under program control (using standard IBM carriage control characters) or under the control of the 1612 Monitor Channels 1-6.

USAGE

1. Operational Procedure

   a. Load the program at location 100. Put the tape to be listed on logical tape handler N.
   b. Put OONP in the A register, where P = 0 for program control, P = 1 - 6 for Monitor Channel 1 - 6.
   c. Run. The first file will be listed, and the program will stop at location 215. If there are more files to be listed, repeat (c). A page eject will occur after each file.

   If it is desired to print several files, and the end of useful information is denoted by 2 successive EOF marks, the only changes needed are to change location CONTIN + 3 (octal 261) to be LDN 0 (0400), and REWIND (octal 215) to be NXF 3 (6103). Then the program will rewind the tape to the unload point and stop at REWIND 5 (octal 222).

3. Space Required

   $412_8 = 266_{10}$

4. Temporary Storage

   $202_8 = 130_{10}$ locations, plus the storage needed for the longest tape the longest tape record.

5. Error Print-outs

   If a line has a parity error in it, the words PARITY ERROR will be printed before the line.

8. Input Tape Mounting
   On logical tape N, when the A register contains OONP.

9. Input Format

A record may have any number of characters up to 7266_{10}. The character 32_8 in an even position is interpreted as an end-of-record mark. This will terminate the line, and the 32_8 will not be printed. If program control is desired, the first character of every line will be interpreted as a carriage control character, and will not be printed. Legal CC characters are 1 (page eject), 12_8 (double space), and 40_8 (triple space). All others are interpreted as single space.

10. Timing

1000 lines per minute, if the blocked records do not exceed approximately 1250_{10} characters.

12. Cautions to User

If blocked records are used, each line image must be ended by a 32_8 in an even character position. If program control is also used, the 32_8 must be followed by a CC character.

13. Equipment Configuration

160 or 160-A, 1612, and 1607.

METHOD

The first record is read into core in assembly mode, starting at INAD(0713). The program then unpacks the information, storing 1 character per word starting at BFINAD (0523). When 32_8 occurs in the right half of an input word, or the end of the input record is reached, the program prints the unpacked characters on the 1612.

F4
IDENTIFICATION

Title: Flexowriter Tape to ANelex - Ident. F4.07 (Flexlex)
Programmer: Bud Vitoff, February 1961

PURPOSE

1. Flexlex was designed primarily for printing OSAP listing tapes. It can be used efficiently whenever the ANelex is available for use immediately after assembly.

2. The routine can be used for printing any paper tape prepared by or for a Flexowriter.

USAGE

1. Operational Procedure

   a. Load program into 0000 through 0650.
   b. Turn on reader. Insert tape to be printed anywhere on its blank leader.
   c. Master clear and start. A page eject is executed before printing starts.

   d. When ten consecutive blank frames are detected, a page eject is executed and printing stops. Starting at this point enters a closed "page eject and stop" loop.
   e. A master clear restarts the program.
   f. A closed "page eject and stop" loop can be entered by setting P = 0001.

7. Programmed Stops

   Z = 7700: Page eject loop entered at 0001
   Z = 7702: Page eject loop entered after ten consecutive blank frames.
   Z = 0004: Program check sum failure. Reload program.

9. Output format

   a. Lines per page: 56
   b. Flexowriter code interpretation (7th level is ignored):
      1. carriage return and tabulator codes cause appropriate printing action, with tab stops after every 12 columns (i.e., in cols. 13, 25, 37, etc.).*
      2. delete codes and illegal codes are ignored.

3. all other non-printing codes are represented by the indicated substitutes:

Code	Meaning	Substitute
02	color shift	'
43	stop	*
47	upper case	:
57	lower case	?
61	backspace	(

c. An automatic carriage return is executed after 120 columns have been "set" for a line of print.

10. Timing
Printing speed varies with the amount to be printed on each line, but OSAP listings run about 325 lines per minute.

11. Equipment configuration
Basic 160 computer and ANelex.

*Tab stops must be equally spaced across the page; however, the routine may be easily changed to provide one of the following options:
1. tab interval fixed at a value other than 12, or
2. tab interval fixed, with provision for changing it at the beginning of each run (by manually changing the contents of register A at a programmed stop).

The tab interval is set at the beginning of a run by the instructions in location 0107 and 0110:

Location	Current	Change 1	Change 2
0107	LDN 14		LDN xx
0110	LDN 14	LDNxx	HLT 01

Load the program, and make the change in core. Master clear and start. Error stop 0004 will display the calculated new check sum in register A.

Permanent change:

Enter the new check sum into location 0650 and punch a new bi-octal tape.

Temporary change:

Start. Program stop 7704 will display the programmed check sum in register A. (Of course, the check sum difference should be accounted for by your changes.) Start to enter the program.

F5.
IDENTIFICATION

Title: Tape-to-Card Punching Routine - Ident. F5.02
Programmers: E Campbell, R. Mills, October 1962, Control Data Corp.

PURPOSE

To read magnetic tape in BCD or binary mode and punch corresponding Hollerith or binary cards.

USAGE

1. Operational Procedure
Mount input tape on a 163 or 164 tape handler. Set the 523 punch unit to ready. Load the bioctal program tape at zero, set the A register to one of the input parameters, set the P register to one of the entry points, and run.

2. Parameters

a. Input - A Register
Load the A with OXYZ where:
X = 1 halt at each end-of-file mark
X = 0 process to end-of-tape
Y = 1 tape unit 163 or 164 - only legal tape unit at present
Z = 1 tape handler number 1
Z = 2 tape handler number 2

b. Entry Points - P Register
Normal entry - P = 100
Read to next end of file and halt - P = 1077

3. Space Required
947 decimal or 1663 octal locations

5. Bank Allocations
When operating this program on the 160-A, be sure all bank settings are the same, e.g., R=0, I=0, D=0, B=0, or R=1, I = 1, D=1, B=1.

6. Alarms or Printouts
This routine was written to process punch output from the 1604 CO-OP Monitor System, consequently the program looks for a job number. No job number is required to use the program, but if there is one it will be displayed in the A register at halt location 324 octal if non-recoverable parity error occurs. This number will be in 4-bit BCD code allowing a maximum job number of 999

A second halt at location 326 occurs with parity errors. A contains the octal punched card count. If processing is continued from this halt, an X card is punched and the parity error record follows. Normal processing will then continue.

7. Error Stops

Location	Cause	Comments
0101	Illegal input parameter	Set A and run
0324	Solid tape parity error	(A) is job number in 4-bit BCD
0326	Solid tape parity error	(A) is nth punched card in octal
0713	Halt at each end-of-file	(A) is job number in 4-bit BCD
1074	Normal halt at end-of-tape	Tape will have been rewound to unload.
1104	Illegal unit number when EOF entry was used	Set A and Run
1135	Halt after skipping one file	Run if processing is desired, reset P and A if next EOF is desired

10. Input/Output Formats

a. Input
Tape records may be 84 characters or less in BCD, binary or a mixture of both modes. Two consecutive end-of-file marks or an END MONITOR OUTPUT card will terminate processing.

b. Output
Hollerith or binary punched cards.

11. Timing
The program operates at punch speed (100 rpm).

14. Equipment Configuration
163 or 164 tape unit
Minimum SWAP 160 or 160-A computer
1610-523 card punch system

METHOD
The program reads magnetic tape in 12-bit word mode (2 frames/word) for binary records and 6-bit word mode (1 frame/word) for BCD records. If a parity error occurs, a reread in the binary mode is tried 3 times, then 3 tries are made in BCD mode. If the parity condition persists, it is classified as a solid parity error. If no parity error occurs the mode continues used until the next parity error.

F6.
IDENTIFICATION

Title: Flex Tape to Magnetic Tape Converter – Ident. F6.01
Programmer: L. Kuller, August 1960

PURPOSE

Produce a copy of a Flexowriter tape on magnetic tape in a form suitable for listing.

USAGE

1. Operational Procedure

   a. Load program tape F6.01

      1. Turn on reader, insert tape and set P = 0000.
      2. Set Load switch and press Run.
      3. First stop
         P = 0004
         A = 1111
         Z = 0000
      4. Set P = 1000, press Run switch
      5. Program Stop
         P = 1177
         A = 0000
         Z = 0000

   b. Convert Flexowriter tape to magnetic tape

      1. Turn on reader and insert Flexowriter tape.
      2. Set CODED parity selection on magnetic tape unit.
      3. Set P = 1000
      4. Press Run switch
      5. Program Stop (16 consecutive blank frames read)
         P = 1021
         A = 0000
         Z = 7707
      6. Option-write end-of-file mark, clear computer and press Run switch.
      7. To convert more Flexowriter tape, return Run switch to center and then to RUN.

3. Space required - 204_8 - 132_{10} locations

7. Error - parity, routine attempts to write output block on tape until the error disappears.

13. Equipment Configuration - Minimum 160 computer with magnetic tape units.

METHOD

Punched paper tape characters are translated into equivalent line printer codes. If there is no equivalence, the punched paper tape character is ignored. When a CR is read, blank codes to fill the 120 character line printer line are inverted on the magnetic tape. The output from the 160 is copies on magnetic tape with parity checking.

F6.
IDENTIFICATION

Title: 1609 Card to Magnetic Tape (1609-164) - Ident. F6.02
Programmer: C. Frankenfield, October 1961

PURPOSE

This routine will read BCD cards, check validity, and write BCD card images on magnetic tape.

USAGE

1. Operational Procedure

   a. Load the program binary tape using the OSAP Binary Loader Program (A4.00).
   b. Place cards in card reader hopper 9-edge first, face-down. Cycle one card. Select tape unit 1.
   c. Set P-register to 6600 and RUN.
   d. Normal Stops

      6635 - Card stacker full. Remove cards from stacker; RUN from this location to continue program.
      6661 - Card hopper empty.
         a. To continue program reload card hopper and RUN from this location.
         b. To read last two cards in reader and write end of file, enter 1 in A-register and RUN from this location.
      7351 - Last two cards have been read and written on magnetic tape. End of file has been written.
         a. To write more card images on same tape, reload card hopper, feed one card, and RUN from this location.
         b. To rewind magnetic tape, enter 1 in A-register and RUN.
      7356 - Tape has been rewound. Depress RUN to perform next job.
      7402 - End of magnetic tape has been reached. End of File has been written and tape has been rewound. After changing magnetic tapes, RUN from this location without removing cards from hopper to continue program.

3. Space Required

   Program occupies locations 6600 through 0000 and low core locations 70 through 76.

7. Error Stops

6642 - Card feed failure. Impossible to recover at this point. Repeat Step b under Operational Procedure and RUN from this location to restart program.

6653 - Card reader not ready. Correct status fault; RUN.

6657 - No cards initially in hopper. Correct fault; RUN.

7174 - Illegal BCD character read.

a. RUN, and card with illegal BCD character will be ignored.

b. Enter 1 in A-register and RUN to write card with illegal character on tape.

7466 - After 3 attempts at three successive locations unable to write a record due to parity errors or illegal BCD code.

a. RUN from here to try to write on magnetic tape again.

b. Enter 1 in A- register, and RUN to start over again with new tape.

10. Timing

The program reads at reader speed (100 cards per minute).

13. Equipment Configuration

Minimum 160 with 164 and 1609 adaptor for the IBM 521 Card Read Punch Unit.

F6.
IDENTIFICATION

Title: CARD TO MAGNETIC TAPE (167 to 163/164) - Ident. F6.03.
Programmer: Anthony Taylor Smith, October 1962, Control Data Corporation

PURPOSE
This program is written in OSAS language for either the 160 or 160-A computer. It converts data from cards into magnetic tape records, one card per record. Either binary or BCD cards will be handled, producing binary or BCD records, respectively. (A binary card is identified by a 7-9 punch in column 1). These cards may be mixed.

USAGE

1. Operational Procedures

The program is distributed as a binary tape and should be loaded with a starting address L between 0100-7410 inclusive; the program will then be run with the starting address, L. in the P register.

Use tape unit 1 for the output tape.

The program will process cards until the card hopper becomes empty, at which time the 'card hopper empty' status response produces a halt at Location L + 61 (L+61 0000 7700). If more cards are to be loaded in the same magnetic tape file, NON-ZERO must be put in the A register before running from this point. To produce an end-of-file mark after the hopper-empty condition, leave ZERO in A and run.

Note: Cards with the same leading corner cut must be used throughout.

3. Space Requirements

The program 247_8 cells which can be loaded anywhere within core locations 0100-7656 and location 70_8 in low core. Locations 7657-7776, inclusive, are used for the 80-column read-in-area, one column per word.

7. Error Stops

L+112 10XX 0000 - XX is the 167 status response code. Clear 167 error condition. Note value of P register, clear, enter noted P plus 1, in P, 0 in A, and run.

L+241 0XXX 0000 - XXX is the card column (in octal) where an illegal code exists. Correct the card, replace in reader, and run.

14. Equipment Configuration

Minimum SWAP 160 or 160-A, 167 reader and a 163 or 164 tape unit.

F6.
IDENTIFICATION

Title: Change Magnetic Tape Record Length Binary Tapes - Ident. F6.04
Programmer: (Control Data Corporation/Applications), October 1962

PURPOSE

Writes magnetic tapes of a designated record length from tapes of the same or different record lengths for binary tapes only (odd parity).

USAGE

1. Operational Procedure:

   Load and run at 0, TU1 input, TU2 output.

   At the halt 0002 0000 7700 enter required output record size in the A register and run.

   Halt 0123 0000 7700 allows two or more input tapes to be output with no "zero fill in" gap between them. At the halt enter either zero or positive non-zero into the A register. In the latter case, a new tape must be readied on TU1-run. To terminate the process make no entry in the A register-run. The program will fill up the last record with 0's to the required size, output it and a tape mark on the output tape.

   Final Halt, 0157 2112 7700: After this halt the program reinitializes itself, so that another process can be handled without reloading the program.

   Clear and run from 0.

   Note: Since the program was originally written for the 160 computer and is at this time designed to be compatible with either the 160 or the 160-A the sum of the output and input block lengths must not exceed 7500_8. Further, in operation on the 160-A all banks should be zero.

3. Space Requirements:

   Program uses locations 0000-0163. From location 200 on is made available as the input/output area.

5. Bank Allocation: Set all banks on the 160-A to zero.

7. Error Stop:

0031 0000 0000: Delinquent record.
The tape has been read three times without success, enter non-zero in A to bypass delinquent record, or 0 to keep trying to read the same record.

14. Equipment Configuration:

Two 163/164 Magnetic Tape Units, 160 or 160-A SWAP Computer.

F6.

IDENTIFICATION
Title: 160-A Binary Card Binary Tape Verify Routine - Ident. AF6.06
Programmer: Larry Brown, Control Data Corporation

PURPOSE

This routine compares 80 column binary cards with their image on magnetic tape. If the card and tape image do not match, a card is punched representing the tape image. The routine has been written for the IBM 533 Card Read Punch.

USAGE

1. Calling Sequence and Operational Procedures

   a. As a closed subroutine, entry is made with the following calling sequence:

```
L       JPR
L+1       VERIFY
L+2     NORMAL
L+3       RETURN
L+4     OWOU
```

   b. As an independent program, load the bi-octal tape at location 0060 in either bank and start at location (r) 0060 with OWOU in the A-register.

   c. 533 Card Read Operation

      Place the cards to be verified in the read hopper to feed 9-edge first, face down. Push the START button twice to feed two cards and turn off the NOT READY light. At least two cards must follow the deck to be read; these cards may be blank or the first two cards of the next deck to be verified if more than one file is on tape.

   d. Card Read Feed Failure

      In case of a card read feed failure, follow these procedures:

      I) Put the read stacker switch in the <u>ON</u> position (toggle it toward you).

      II) Remove the cards from the read hopper. Reproduce the damaged cards from the bottom of the stack and discard the damaged cards. Put the reproduced cards at the bottom of the stack.

III) Push the START button until all cards have been fed out of the machine.

IV) Put the last card fed out in front of the reproduced cards from the read hopper.

V) Place the corrected card stack in the read hopper and push the start button twice to feed two cards.

VI) Put the read stacker switch in the OFF position (toggle it away from you). The NOT READY light will go out and the computer will continue reading cards.

e. 533 Card Punch Operation

Place blank cards to feed 9-edge first, face down, in the card punch hopper. Push the START ubtton once to feed a card and turn off the NOT READY light.

f. Card Punch Feed Failure

In case of a card punch feed failure, remove the blank cards from the hopper and examine the cards on the bottom of the stack. Discard all damaged cards and replace the stack in the hopper. Push the START button once to turn off the NOT READY light and place a card in position for punching.

g. Procedures for NOT-COMPARE Stop

If the binary card read in does not match with the binary tape image, the computer punches a card and stops.

I) Remove the blank cards from the card punch hopper and feed all cards out. There will be 2 cards, the last one of which is blank and should be placed with the other blank cards. The first card out is the card punched from the tape image.

II) Remove the cards from the card read hopper and feed all cards out of the machine. The last two cards fed out are to be placed in front of the cards from the read hopper. The third last card fed out is the one which did not compare with the tape image; discard that card.

III) Put the card punched in front of the stack from the read hopper (in front of the last two cards fed out in the last step).

IV) Place the corrected deck in the card read hopper and push the START button twice to feed two cards and turn off the NOT READY

light. Place the blank cards in the card punch hopper and push the START button once to feed one card and turn off the NOT READY light.

V) Place the computer's RUN-STEP switch into the STEP position and then into the RUN position. Computer operations will then continue.

2. Entrance Parameter

OWOU as follows:

W = tape control unit number (usually 1)
U = logical tape number selection

3. Space Requirements

325_8 locations in the relative bank
5_8 locations in the direct bank
374_8 locations in the indirect bank

Locations in the indirect bank are assigned by EQU cards, 124_8 - 84_{10} non-overlapping locations must be reserved for the symbols BLKA, BLKB, and BLKC. Location (i) 0000 must not be included in this storage assignment. If it is desired to re-arrange the direct storage assignments, only the symbol SETS may be assigned location (d) 0000. All other direct storage locations may be assigned at will.

7. Program Stops

Location	Description
CKPAR+7	Three attempts have been made to read a tape record and all have failed because of parity errors. A card has not been read for this tape image. Continuing from this point tries three more times to read the record. This stop is at (r) 0223 when used as an independent program.
BBLKB+1	This stop is the NOT-COMPARE stop. A new card has been punched from the tape image. Follow procedures outlined in C.1.g. above. This stop is at location (r) 0346 when used as an independent program.
RWD+1	An End-Of-Tape reflective spot has been encountered. The program rewind-unloads the current tape. Mount the next tape anc continue. This stop is at location (r) 0375 when used as an independent program.

If this routine is used as an independent program, one additional stop is provided: at location (r) 0063 for the normal return.

8. Input Tape Mounting

The input tape is mounted on control unit W with logical selection U as in 2. above. The tape may be file protected.

9. Input and Output Formats

a. Card Input

Standard 80 column binary card.

b. Tape Input

160_{10} character column binary image in odd parity of card 9.a. above.

c. Card Output

Standard 80 column binary card of image 9.b. above.

10. Timing

Approximately 200 cards per minute compared. This routine operates as fast as the IBM 533 read feed.

12. Cautions

Do not step through the routine or take the computer out of RUN.

13. Equipment Configuration

Minimum 160-A with 163 and 1610 adaptor for the IBM card read punch. The program assumes a 2-bank 160-A. The 163 tape unit is selected in odd parity and assembly (12-bit) mode only. The buffer channel is used to read cards from the second read station. The normal channel is used to read tapes and punch cards.

15. Miscellaneous

a. Normal Return

Follows reading an End-of-File mark.

b. Interrupt

Not used.

c. Bank Settings

Upon normal return, the direct, indirect and buffer banks are restored to their values as the time of entry to the subroutine. The subroutine will work in either bank.

d. Machine Registers

The Buffer Entrance, Buffer Exit, and A-registers are not restored.

e. Selective Stop and Selective Jump switches are not used.

F7.
IDENTIFICATION

Title: FORTRAN cards to Flex Tape - Ident. AF 7.00
Programmer: N. D. Babic, August 1962, Control Data Corporation

PURPOSE

This routine reads FORTRAN source cards using the 167 Reader and punches paper tape acceptable to the FORTRAN compiler, eliminating blank columns and inserting Tab, Carriage Return, and Case Code characters as applicable.

RESTRICTIONS

1. Comments cards (C in column 1) are not punched on flex tape.
2. Only blank columns in a Hollerith field are punched as semicolon.
3. Card reader assumed to be on normal channel.
4. Both 11 and 8-4 punches are converted to minus sign.
5. In accordance with existing FORTRAN restrictions, format statements must be the first statement in a program.
6. Blank card terminates the run.

USAGE

1. Operational Procedure
   a. Machine load the program at P = 0000.
   b. Master clear.
   c. Ready FORTRAN deck followed by three blank cards in reader.
   d. Press RUN switch.
2. Programmed Halts

P setting	Cause
$(0216)_8$	End of run - for next run start at 1(b) in Oper. Proc.
$(0162)_8$	Err - number of characters in Hollerith field not specified
$(0406)_8$	Card reader - program error, amplifier failure, or motor power off. Take appropriate action; to continue set P = 0375 and press RUN switch.
$(0407)_8$	Feed failure; to continue set P - 0375 and press RUN switch.
$(0410)_8$	Stacker full; empty stacker, set P = 0375 and press RUN switch.
$(0411)_8$	Hopper empty; place additional cards in hopper, set P = 0375 and press RUN switch.

3. Space Required
   Program occupies locations 0000 through 1020.
10. Standard FORTRAN input and output formats.
11. Timing - None available.
14. Minimum swap 160-A computer plus 167 Card Reader.

F8.
IDENTIFICATION

Title: Plot from Magnetic Tape 163/164 to 165 Plotter via 160-A Computer - Ident AF8.01
Programmer: (Control Data Corporation/ Applications), August 1962

PURPOSE

This routine, written in OSAS-A, is designed to plot consecutive points on the 165 plotter and connect them with the best straight line. Points are obtained from magnetic tape, one point per two-word assembly-mode record, x then y. The first point obtained is used as a base reference point and the plot proceeds relative to this point. There are amny switch options, including a library of graph identification marks (upright cross, diagonal cross, square, circle, diamond, eight triangles and two starts) under the control of Jump Switch 1. Tapes must be in binary mode only (odd parity).

USAGE

1. Operational Procedures:

The first record on the program paper tape is the OSAS-A paper tape loader. Load anywhere in any bank and run from the load address with the desired starting address for the routine in the A register. Run the routine from its starting address. There will be a halt (L+10 0000 7700) at which the tape unit being used should be put in the A register. At a second halt (L+32 0000 7777) selection may be made, by a suitable entry in the A register (0 through 17), from the library of graph marks to identify different plots.

Selective Jump Switch 1 Up Produces a graph with each point marked by the previously selected graph symbol. Without the switch Up the plot proceeds smoothly but the individual points comprising the plot will not necessarily be evident.

Stop Switch 1 is used to effect a pause (L+240 0000 7700); running will allow continuation of the same plot.

Jump Switch 2 is used to effect an absolute stop (L+153 1161 7700) i.e., to change tape reels or tape units. Running from here requires a new selection of a tape unit at the initial halt. This switch setting produces a rewind to load point of the selected tape unit. This switch setting also produces a resetting of all the bank controls to their initial values.

After a tape mark is read Jump Switch 4 is interrogated: Up, the pen stays where it is; Down, the pen is returned to the starting point for the plot, and then a program halt occurs, L+230 0000 7700. Zero in the A register allows a new plot to proceed from the original base reference point after selecting from the graph mark library at the next halt, L+32 0000 7700. Non-zero in the A register continues the previous plot.

3. Space Requirements

Total space requirements for the routine are 1264_8 locations anywhere in any bank. Approximately half of the routine (the latter 526_8 locations) are involved with the tape identification marks and could be dispensed with if it is not required to mark the plots.

7. Error Stops:

L+111 4444 0000 Indicates a suspicious record on magnetic tape. Tape has been read three times and each time a parity error has occurred. Putting zero in the A register allows the bad record to be ignored; running without disturbing A allows reading the tape three more times. This may be repeated indefinitely.

14. Equipment Configuration:

160-A Swap Computer, Tape unit 163/164, and 165 Plotter.

F8.
IDENTIFICATION

Title: 165 Alphanumeric and Special Character Demonstration Plot - Ident. F8.02
Programmer: (Control Data Corporation/Applications), July 1962

PURPOSE

This routine, written in OSAS for the Control Data 160 or 160-A computer, is basically a demonstration routine, but it may be used to produce a neat labelling of plots on the 165 plotter. The basic letter size is six-tenths of an inch and all letters or digits with curved portions are produced with smooth curves, utilizing the small internal up and down, sideways or diagonal steps of the plotter to the limit. The program works from either paper tape or the console typewriter.

USAGE

1. Operational Procedures:

   a. 160 - Load and run anywhere in core. At the halt (L+1 0000 7710) entering non-zero in A allows input from paper tape, otherwise input is via the console typewriter. (More than one character at a time can be typed, but care must be taken not to overload the typewriter buffer since this will result in characters being lost.)

   b. 160-A - Load and run anywhere in core with care that the indirect and relative banks are the same bank. Selective Jump Switch 1 UP allows input from paper tape, otherwise input is via the console typewriter.

3. Space Requirements: 4504_8 locations

7. Error Stop:

   In either computer an error stop L+13 7777 0000 indicates that the 161 typewriter is not ready for the entering of characters. Correction os the condition before running will ensure success.

12. Caution to Users:

   Since the routine is written for the 160-A, the selective jump instruction is a two-instruction set, which, on the 160, will be regarded as individual instructions.

   The second instruction of the two-instruction set will be an error halt

if, and only if, the routine is loaded in 0; otherwise, garbage.

Therefore, for the 160:

a. The routine must be loaded at 0, and

b. A normal error halt will occur after the first normal halt. (Error halt will occur at L 2 after the normal halt L+1).

Note: This error halt does not constitute an error fault.

14. Equipment Configuration: Minimum 160 or 160-A SWAP Computer, 165 Plotter.

G1.
IDENTIFICATION

Title: TRACK - Ident. - G1.00
Programmer: R. Beale, March 1961

PURPOSE

Trace a program, providing a flex-coded paper tape as output. Only the beginning and ending addresses of a consecutive instruction string appear as output, thus the object program is executed at higher speed than is possible using a full trace.

USAGE

1. Operational Procedure

   a. Clear memory
   b. Machine load the bioctal tape starting at 7000 - correct loading will end with P = 7577.
   c. Machine load the object program without altering locations 7000 - 7600_8. Position input data tape in paper tape reader if required.
   d. Set P = 7000, A = starting address of program to be traced. Run. Halt 7701 will immediately occur - P = 7002. Without otherwise altering console, set A = normal contents of A at start of object program. Turn on punch, run.

3. Space required

   7000 through 7534
   $535_8 = 349_8$

10. Timing

    Depending on the nature of the object program, its instructions are executed at 10-500 per second. The average is close to 75 per second.

12. Cautions to User

    Because each instruction must be interpreted before execution, the timing relationships within a program are altered - it is not possible to trace most card to tape programs for this reason. The track program attempts to faithfully execute a sequence of instructions regardless of its correctness. If proper selection of peripheral equipment does not precede the activation of the equipment, the computer hangs on a "sel" error indication. Punching by the track program does not alter the object program external function selections or senses. Each instruction is executed from upper cord, rather than from its normal position memory; therefore programmed error stops or halts in the object program show P = 7146.

H1.

IDENTIFICATION

Title: Binary to 4-bit Decimal Conversion - Ident. H1.00
Programmer: Sandord Elkin, March 1961

PURPOSE

This subroutine will convert a 24 bit binary integer into a decimal integer with each digit in successive cells.

USAGE

1. Operational Procedure

   The 24 bit integer must be placed in locations BINDEC + 100_8 and BINDEC + 101_8, and the return address in cell 6. The 8-digit answer will be put in cells 70_8-77_8, with the units digit in 77_8.

3. Space Required: $67_{10} = 103_8$ locations.

4. Temporary Storage: Octal locations 6 and 70-77.

11. Timing = Approximately 4.3 milliseconds per decimal digit.

14. Equipment Configuration: Minimum SWAP 160 computer.

MATHEMATICAL METHOD

The binary integer is divided by 12_8. The remainder is stored in the appropriate location and the quotient used as a new binary integer.

H1.
IDENTIFICATION

Title: Convert Binary Coded Decimal to Binary - Ident H1.01
Programmer: J. Pedersen, August 1960

PURPOSE

This subroutine will convert a binary coded decimal number of up to six digits to the equivalent binary number in 22 bit arithmetic format. The binary coded decimal number is stored one digit per word with the digit as the low order four bits of the word.

USAGE

1. Operational Procedure

   The address of the high order digits of the decimal number is specified by the contents of storage location 0010. The number of digits in the decimal number is specified by the contents of storage location 011. The resulting binary number will be stored in location 0012 and 0013 with the high order portion of the word in 0012.

   The 22 bit arithmetic format uses the high order bit of the low order word as a buffer to catch overflows, thus this bit must be a zero and is not considered as an information bit. The low order bit of the high order word will be considered as the 2^{11} bit.

   On completion, the contents of location 10 will point one location beyond the low order digit of the decimal number. Contents of 0011 will be unchanged. The routine uses the high order four bits of location 0013 to catch the overflow from the multiply process. This overflow is added to the low order four bits of 0012. On completion of the routines, the program reassembles words 0012 and 0013 to the 22 bit arithmetic format. The assumption in the conversion routine is that the numbers are positive. Any sign indication will have to be added later.

2. Argument or Parameters - None

3. Space Required - 37 locations

4. Temporary Storage Requirements

   0010 - Contains address of high order digit of decimal number
   0011 - Contains count of number of digits in decimal number
   0012 - High order binary result
   0013 - Low order binary result

0070 - Counter used in routine
0077 - Mask (0377)

10. Timing - The routine takes 230.4+ 198.4 N microseconds where N is the number of decimal digits to be converted. The routine will take approximately 1.2 milliseconds to convert a five decimal digit number. The time to convert binary coded decimal information to the corresponding binary information is given in the table below. These times are derived on the assumption that the binary coded decimal information is stored one digit per word with the high order digit of the number given first.

The break in the time sequence between 3 and 4 digits, 6 and 7 digits, 9 and 10 digits is based on changing from a single to double to triple to quadruple precision binary representation.

CONVERSION TIME

Number of decimal digits	Conversion time in milliseconds
2	0.11
3	0.18
4	1.03
5	1.23
6	1.43
7	2.6
8	2.9
9	3.4
10	5.0

11. Accuracy - The routine is good for up to six decimal digit numbers.

13. Equipment Configuration - Minimum 160 Computer.

H1.

IDENTIFICATION

Title: 160-A Output - Ident. AH1.02
Programmer: D. T. Miller, January 1962

PURPOSE

To set up a 120 character line image for on-line printing or writing on a BCD output tape. The format will be specified in the calling sequence.

USAGE

1. Calling Sequence

The routine is entered by a return jump. The calling sequence will be of the following form:

```
L    JPR
L+1  OUTPUT
L+2  NORMAL
L+3  RETURN
L+4  ERROR
L+5  RETURN
L+6  ADDRESS OF ERROR
L+7  XXXX  ┐
L+8  XXXX  }     Format Specification Slot - as many as desired
L+9  XXXX  ┘
 .
 .
 .
L+N  XXXX    ┐
L+N+1  XXXX  }   Output Mode Specification Slot - as many as desired
L+N+2  0000  ┘
```

The calling sequence must be in the same bank as the output program. Three full words in an output specification specify a field in the line image or a mode of output. The format specification may refer to data in any bank. Although ordering within the format specifications is immaterial, the entire group of format specifications must precede the first output mode specification word. At least one output mode specification must be given.

2. Options and Parameters for Format Specifications

a. Binary-to-Octal

```
L    OPPP
L+1  DATA
L+2  NOOB
```

PPP: The rightmost print position to be used for this field.
DATA: The base address of the data word.
B: Bank number of the bank the data word is in.
N: The number of octal digits for output; if less than four octal digits, the octal digits should be right justified. The maximum is four digits.

b. Hollerith-to-Hollerith

```
L    1PPP
L+1  DATA
L+2  NNOB
```

PPP: The rightmost print position to be used for this field.
DATA: The base address of the data word.
B: Bank number of the bank the data word is in.
NN: Number of words in the BCD field.

The BCD information is transferred in whole words, any blanks in a BCD field must be provided by the programmer. The maximum number of words is 60 (120 characters).

c. Integer-to-Integer

```
L    2PPP
L+1  DATA
L+2  000B
```

PPP: The rightmost print position to be used for this field.
DATA: The base address of the data word.
B: Bank number of the bank the data word is in.

The output is 5 characters, sign plus 4 decimal digits. The integer is converted by successive subtraction of powers of ten.

3. Options and Parameters for Output Mode Specifications

a. Print On-Line

```
L    4000
L+1  00CC
L+2  0000
```

CC: Carriage control character, any of those applicable to the CDC 1612 printer. The paper motion will be pre-skip except for selection of one of the six monitor channels, then it will be post-skip.

b. Write on BCD Tape

```
L    3U0T
L+1  K0CC
L+2  0000
```

U: Tape control unit number used.
T: Tape number in proper unit.
CC: Carriage control character, any of those applicable to the CDC 1612 printer. The paper motion will be pre-skip except for selection of one of the six monitor channels, then it will be post-skip.
K: K = 4 provides for blocked output. The line image is stored in blocked form until a normal write tape sequence is encountered, and then outputs the accumulated lines in one record. A maximum of 5 lines may be blocked, then they will be put out in one record and blocking will continue.

The routine is coded for the CDC 163 tape unit.

6. Error Returns

In the event of error, error indications are placed in the A-register and the calling sequence. The address of the erroneous calling sequence word is stored in the calling sequence to the output routine. One of the following numeric codes is placed in the A-register:

1 = Print position not in the range $0\text{-}120_{10}$.
2 = Illegal output specification number.
3 = No output mode specification given.

8. Tape Mounting

Controlled by programmer.

9. Format of Input and Output Data

a. Numeric data is assumed to occupy one full word in memory. Hollerith data is assumed to occupy one or more consecutive full words in memory for any one calling sequence reference.

b. Output: Controlled by programmer.

10. Selective Jump and Stop Settings

None.

11. Accuracy

Conversions are designed for full significance obtainable.

12. Cautions to the User

a. No check is made for overlapping fields. Any entry to the image supercedes any previous entries in the same position (the image is initially filled with BCD blanks).

b. The routine does not interfere with any buffering in operation at the time of entry.

c. No tape checking is done within the routine.

13. Total Space Required

618_{10} Program. 305_{10} temporary storage for blocked records.

H1.
IDENTIFICATION

Title: Conflex - Ident. AH1.03

PURPOSE

This routine allows you to use the *Flexowriter Model 35-4 in the 6 level mode to prepare programs and data for the 160A. All 35-4 Flexowriter characters produce equivalent 160A Flexowriter characters except ", $, Δ, @, #, ϒ, *, _, ?, and uppercase period.

OPERATING PROCEDURE

1) Master clear.
2) Set all banks equal to 1
3) Mount conflex
4) Machine load at P = 0000
5) Set P = 0140
6) Mount tape prepared on 35-4 Flexowriter
7) Turn on punch
8) Run
9) Operation is terminated by moving run switch to off.

*(Standard Flexowriter used with G-15)

Note: If jump switch #1 - ON computer will stop on illegal character.
If jump switch #1 - OFF computer will not stop on illegal character.

III. Character Table

LC	UC	G-15 6 Level Flex	160A Flex	UC	LC	Notes
a	A	014	30	A	a	
b	B	034	23	B	b	
c	C	002	16	C	c	
d	D	022	22	D	d	
e	E	012	20	E	e	
f	F	032	26	F	f	
g	G	006	13	G	g	
h	H	026	05	H	h	
i	I	016	14	I	i	
j	J	036	32	J	j	
k	K	001	36	K	k	
l	L	021	11	L	l	
m	M	011	07	M	m	
n	N	031	06	N	n	
o	O	005	03	O	o	
p	P	025	15	P	p	
q	Q	015	35	Q	q	
r	R	035	12	R	r	
s	S	003	24	S	s	
t	T	023	01	T	t	
u	U	013	34	U	u	

LC	UC	G-15 6 Level Flex	1600 Flex	UC	LC	Notes
v	V	033	17	V	v	
w	W	007	31	W	w	
x	X	027	27	X	x	
y	Y	017	25	Y	y	
z	Z	037	21	Z	z	
2	"	111	70	----	2	
3	'	131	64	See Note	3	Punches 44 which is '
4	$	105	62	----	4	
5	π	125	66	----	5	
6	@	115	72	----	6	
7	∅	135	60	----	7	
8	△	103	33	----	8	
9	(	123	37	See Note	9	Punches 54, which is (
0.	)	101	56	See Note	0	Punches 57, 54, 47 which is)
1	*	121	74	----	1	
Stop 2	Stop 2	104	43	Stop	Stop	
BS	BS	132	61	BS	BS	
tab	tab	130	51	tab	tab	
CR	CR	110	45	CR	CR	
,	o	126	56	o	See Note	Punches 46 which is ,
.	.	114	42	----		
/	?	124	44	----	/	
----	UC	122	47	UC	----	

LC	UC	G-15 6 Level Flex	160A Flex	UC	LC	Notes
LC	----	102	57	----	LC	
Space	Space	100	04	Space	Space	
Code Delete	Code Delete	137	77	Code Delete	Code Delete	
+	=	136	42	=	See Note	Punches 47, 46, 57 which is +
;	:	116	50	:	;	
-	----	120	52	----	-	

H1.
IDENTIFICATION

Title: General Binary to BCD, BCD to Binary Conversion
FMTCNV - Ident. AH1.04
Programmer: A. R. Perro, October 1962

PURPOSE

Uses a format statement, similar to Fortran statements, to specify the type of conversion to be made in the input or output field. The input and output fields are always internal information and are not in any way associated with external equipment. All input and output information is handled internally and is operated on specifically by the requirements made in the format statement. The format statements are conversion specifications which control the conversion of data from internal machine representation to packed external representation.

FMTCNV is a closed subroutine entered by a return jump (JPR) with the contents of the A-register specifying the type of conversion.

USAGE

1. Calling Sequence:

L-1	LDN	0 or 1	0 = BCD to Binary 1 = Binary to BCD
L	JPR	GENBCD	
L + 2	Location of format statement		
L + 3	Starting Location of Input		
L + 4	Terminating location in input		
L + 5	Starting location of output		
L + 6	Terminating location of output		
L + 7	Error indicator		
L + 10	Return		

3. Space required: $2063_8 = 1075_{10}$ locations

4. Temporary Storage: Direct cells 70-74

5. Bank Allocation: Relative and direct must be the same, indirect must be the same as the main program, and input, output, and format fields must be in the same bank as the indirect bank setting.

8. Error Returns:

An error sets an error indicator at L+7; will also be in the A-register at the time control transfers to the main program.

The error codes are:

A = 1 The first left parenthesis is missing
= 2 Illegal character where a number should appear
= 3 Illegal code for parenthesis
= 4 Illegal imbedded parenthesis
= 5 Digit count exceeds 99 or output exceeds the range
= 6 Comma following a repeat digit or illegal alpha character
= 7 Output exceeded during alphanumeric format
= 10 Integer exceeds range of $\pm 2047_{10}$.

10. Input-Output Formats:

a. Format Statement

The format statement must <u>always</u> be packed BCD information with a left parenthesis beginning the statement and terminating with a right parenthesis. Between the left and right parentheses repeat parens may be used such as:

(2(16,3HABC,4A6),120,2X,4(o5,o2)).

b. Conversion Types

There are two numerical conversion type codes: I, for integer, and O for octal; and three alphanumerical conversion type codes: A, X, and H.

<u>I Conversion</u>

In: Where n is an unsigned integer specifying the number of characters in the field.

Input: Causes conversion between external BCD to internal binary representation. Spaces are entered as zeros and the integer variable may take on values from -2047 to 2047 inclusive. A minus sign will count as a character read. Maximum integer would be 14 for positive values and 15 for negative values. If n is greater than 4 (or 5 for negative) during input, the leading characters of the input field will be ignored and only one work will be stored in the output field. If the integer exceeds the range of ± 2047, the routine will exit to an error indicator.

Example: Input field contains 3 words 0305, 0212, 0407. The result in the output field would be one word 3777_8 and conversion would continue from the character following the 07.

Output: Causes conversion between internal fixed point representation and external BCD representation. The external representation is of the form, (-) DDDD, where D is a decimal digit and the minus sign or blank may appear. The field width must be large enough to include the sign; it may be larger to include leading blanks in the field. Leading zeros are replaced by blanks except for a zero value. Figure 1 shows the effect of various width fields.

15	14	13	12	11
0	0	0	0	0
5	5	5	5	5
86	86	86	86	8
743	743	743	74	7
1395	1395	139	13	1
- 6	- 6	- 6	-6	-
- 23	- 23	-23	-2	-
- 456	-456	-45	-4	-
-1789	-178	-17	-1	-

Figure 1.

O Conversion

On: Where n is an unsigned integer specifying the number of characters in the field.

Input: During input is is expected to find only spaces, digits from 0-7 and possible a minus sign. All other characters are illegal. If n is greater than 4 will cause n-4 leading characters to be ignored.

Output: O specification causes output of the octal equivalent of the 12-bit representation of the integer value. Leading zeros are not suppressed. If n is less than 4, n leading digits will be placed into the output field. If n is greater than 4, n-4 leading spaces will precede the 4 digits in the output field.

The three alphanumeric conversion specifications provide for transmission of alphanumeric field, spacing by skipping character positions, and Hollerith information from the input list. These are converted by A, X, and H specifications.

A Conversion

An: Where n is an unsigned integer specifying the number of characters in the field.
Input: Transfers n characters to the output field. If n is less than 2, n characters are transferred. If n is greater than 2, n-2 leading characters are ignored.
Output: Transfers n characters to the output field. If n is less than 2, n characters plus the necessary leading blanks are transferred to the output field. If n is greater than 2, n-2 blanks precede the alphanumeric characters.

X Conversion

nX Where x is an unsigned integer specifying the number of characters in the field.
Input: n characters are skipped in the input field.
Output: n spaces are placed into the output field.

H Conversion

nH Where n is an unsigned integer specifying the number of characters in the field.
Input: n characters are skipped in the input field.
Output: Causes n alphanumeric characters immediately to the right of the H to be transferred to the output field.

Repeated Groups

A repeat factor may precede any of the K, O and A specification codes. Limits of the repeat are n = 99. Groups may be repeated in a similar manner. The group to be repeated is enclosed in parentheses and a repeat factor given preceding the left parenthesis.

Example: 2(16,3X,4A2)

12. Caution to User: A repeated group may not contain a repeated group.
Example, illegal: 2(16,3X,2(4A2))

14. Equipment Configuration: Minimum SWAP 160-A computer.

H2.
IDENTIFICATION

Title: Sort 3X - Ident. H2.01
Programmer: (Control Data Corporation Sales), November 1962

PURPOSE

Sort 3X is a revised version of Sort 34 which is replaced. Sort 3X accepts input of a file from one reel of magnetic tape where the information is recorded as one item per physical tape block, This information is sorted on up to ten sort key fields using any standard Control Data tape transport. Four tape units are required. The sorted file is written out on tape with one teim per block of magnetic tape. Either a minimum Control Data 160 or 160-A computer is used.

USAGE

1. Operational Procedure

   a. Load Program Tape
      1. Turn on paper tape reader.
      2. For 160-A set all bank controls to bank 0.
      3. Depress the load-clear switch to clear and then place in load position.
      4. Run. Tape will load and stop with P = 3700 A=2361. This loads the program version for the 163, 164, or 162 tape systems.
      5. For 1607 or 1615 tape systems, Clear. Place the load-clear switch in the load position.
      6. Set P = 3360, Run. Tape will load the 1607/1615 tape driver routines and stop with P = 3722 A = 7012.

   b. Define Sort Fields
      1. Place enter-sweep switch at ENTER. Press load-clear switch.
      2. Set P = 0100
      3. Enter octal equivalent of location of left-most character of sort field in Z.
      4. Press step switch.
      5. Enter octal equivalent of location of right-most character of sort field in Z.
      6. Press step switch.
      7. Repeat steps 3 to 6 for each succeeding minor sort field.
      8. Press step switch 2 times to place zeros after last field definition if 9 or fewer fields are used.
      9. Return all switches to center position and press load-clear.

EXAMPLE: Sort a card record on columns 10 to 19 as major field and columns 1 to 5 as a minor field. The 6 numbers entered in the program are in order: 0012, 0023, 0001, 0005, 0000, 0000.

c. Define Sort Order for Magnetic Tape Character Code

Sort 3X includes a sort order table for standard BCD magnetic tape codes. If a different code is to be used, place the sort order table in locations 0200 to 0277 either manually from the console, or by machine loading a previously created binary input paper tape sort order table.

The present sort order from low to high is as follows:
Blank, 0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j k l m n o p q r s t
u v w x y z - & / , % $ * . # @
other special codes

To define a new sort order, for instance, blank A-Z, 0-9, it is necessary to place a sort order number, starting with octal 00 in the location corresponding to the magnetic tape input coding of the required character. Thus, blank comes in as octal 20, a is 61, B is 62, etc. This new sort order would require that location 0220 contain a zero, 0261 contain 0001, 0262 contain 0002, etc.

d. Mount Magnetic Tapes

1. Mount input tape on tape unit number 1.
2. Mount scratch tapes of sufficient length on tape units 2, 3, 4.
3. Manually position all 4 tapes at load point.

e. Running the Sort Program

1. Specify parity of input tape by placing in location 0002 the number 0000 for even parity 0001 for odd parity.
2. Press the load-clear switch on console.
3. Put run-step switch in run position.
4. Input pass will take place. On completion, all tapes will rewind. Tape number 1 will rewind to unload position and computer will stop with P = 0643 A = 0643, Z = 7700. For other stops and corrective action, see Halt Table (Appdx I).
5. Replace input tape on tape unit 1 with a scratch tape.
6. Move run-step switch to the center position and return it to run position.
7. The sort will continue to completion and computer will stop either with P = 0701 A = 0001 or P = 0711 A = 0003. The contents of A will indicate which tape contains the output sorted file. This tape will be in the unload condition.

8. Return run-step switch to center position.

9a. For 162, 163, or 164 tape systems: If the output is on tape unit 1, press load-clear and return it to center. Set P = 0130 and run to rewind the error tape number 2. If the output is on tape unit 3, press load-clear and return to center position. Set P=0135 and run to rewind error tape unit 4.

9b. For 1607 or 1615 tape system: If output is on tape unit 1, press load-clear and return to center. Set P = 3711 and run to rewind error tape unit 2. If output is on tape unit 3, press load-clear and return to center; set P = 3714 and run to rewind error tape unit 4.

f. List Error Tape

The error tape contains records involved in an unrecoverable read situation. These records are also on the sorted output tape. The operator should check and correct such errors.

g. To Perform Another Sort

The sort program is self-restoring and may be used without reloading. If the same set of sort fields is to be used, mount the new input and run the sort program starting with Step e.

h. Rerun Points or Sort Termination

1. To terminate sort (160-A) set optional stop one ON. Program will stop after completion rewind of tapes at end of current merge pass with A = abcd. Octal digits a and b are the tapes which were input to the merge, c, and d, are the tapes with are the output of the merge pass.

2. For a rerun point, remove tapes a and b and save. Replace them with the new scratch tapes. Run from the point of the stop.

3. To terminate sort for further running at a later date, remove tapes 3 and d and save them.

i. Resuming Terminated Sort or Rerun

1. Load sort 3X program, or if already in machine go to step 2.
2. Place intermediate tapes obtained from "h" avice on tape units 1 and 2.
3. Master clear. Place 1234 in A register. Set P = 0764. Step 3 times; this places the correct tape numbers in the read control.
4. Master clear. Set P = 0777 and run.
5. Stops may occur with P = : 1036 1067 1077 1104, in all cases run to continue.

5. Bank Allocation: Set all 160-A bank controls to Bank 0.

7. Error Stops

Appendix I, HALT TABLE, gives each stop and corrective action. Sort 3X provides error checking at several levels. The main source of possible errors is the magnetic tape system; Sort 3X provides checks and corrective action. The checks are as follows:

a. Checks for short records from input tape: Sort 3X places a lower limit of 16 characters for an acceptable input record. If a smaller record is introduced a 0426 stop occurs. If the short record is not expected, it should be discarded from the sort as a probable noise record generated by bad tape.

b. Intermediate tape records must have a given format. Sort 3X checks each intermediate tape record for the correct format before releasing it to the merge operation. Stop 3316 provides the option to try to read the record again or to eliminate it from the merging operation. On elimination, a copy of the record is produced on punched paper tape for further analysis by maintenance personnel.

9. Input and Output Tape Mountings

See 1.d. page 2

11. Timing

To estimate the time required by Sort 3X to process a file of records, it is necessary to know the duration of phase 1, the duration of each merge pass of phase 2 and the duration of phase 3 as well as the tape rewind time required for each pass.

The following definitions and timing formula give the sort time using Sort 34 using various tape units.

N	Number of records to sort
R	Number of characters in record. If odd, take next even number.
K	Number of characters in key, if odd, take next even number.
BF	First pass blocking factor = $\frac{2970}{K+R+4}$ Integer portion of division
P	Number of passes = $\log_2 \frac{N}{BF}$
G	Grouping in merge pass = $\frac{986}{K+R+4}$
NB	Number of data blocks in merge pass $\frac{N}{G}$
BS	Block size for merge in characters = $6 + G(K+R+4)$
CPR	Characters contained per full reel of tape
IBS	Number of characters used for interblock space

TR Transfer rate in thousands of characters per second
RS Read start stop time (milliseconds)
WS Write start stop time (milliseconds)
SD Stop delay (milleseconds dead time for continuous read or write)
RW Rewind time in minutes for a full reel of tape.
MG Internal processing time to merge one group of data to get an output block. (milleseconds)
T Sort time in minutes

The sorting formula for Sort 3X is as follows.

$$T = \frac{(N + \frac{2N}{BF})(RS + SD + \frac{R}{TR}) + (P)(NB)(RS + WS + MG + \frac{2BS}{TR}) + (N(WS + SD + \frac{R}{TR}))}{60,000}$$

$$+ \quad \frac{2N(R + IBS)}{} \quad + \quad (P)\ (.75)\ \frac{(NB(BS + IBS))}{CPR} \quad RW$$

Where $MG = (G)\ (.15)(K + .3R + 12)$

EXAMPLE

This example illustrates the use of the Sort 3X timing formula. Assume that a file of records to be sorted has the following specifications:

Record length (R) = 80
Number of records (N) = 20,000
Length of key (K) = 16 in two fields
163 magnetic tapes with a 30 KC transfer rate are to be used.
First determine the first pass blocking factor $BF = \frac{2970}{K+R+4} = 29.7$

Taking the integer portion we have BF = 29
We then determine the number of passes (P) by finding N/BF = 689
And take the log to the base 2, or use table 2 to determine that P = 10
We next determine $G = \frac{986}{100} = 9.86$ so take G = 9

We then determine $NB = \frac{N}{G} = \frac{20,000}{9} = 2,222$

And determine $BS = 6 + G(K+R+4) = 6 + 9(100) = 906$

Taking values of CPR, IBS, TR, RS, WS, SD, RW from the tape characteristics table and computing $MG = (G)\ (.15)(K+.3R+12) = 70.2$ milliseconds. We substitute in the formula to find the sort time is

$$T = \frac{(20{,}000 + 40{,}000)(4 + 10 + \frac{80}{30})}{60{,}000} + \frac{10 \times 2{,}222(4 + 18 + 70.2 + \frac{1812}{30})}{60{,}000}$$

$$+ \frac{20{,}000(18 + 10 + \frac{80}{30})}{60{,}000} + \left[\frac{40{,}000\ (230)}{5{,}520{,}000} + (10)(.75)\ \frac{2{,}222(1056)}{5{,}520{,}000}\right] 3.33$$

$$T = \frac{(20 + \frac{40}{29})(16.67)}{60} + \frac{10 \times 2.222(152.6)}{60} + \frac{(20)(30.33)}{60} + \left[\frac{4 \times 2.3}{5.52} + \frac{7.5 \times 2.2 \times 1.056}{5.52}\right] 3.33$$

$T = 5.95 + 56.5 + 10.3 + 16 = 88.8$ minutes

Table 2 Number of merge passes

N/BF	P
1-2	1
3-4	2
5-8	3
9-16	4
17-32	5
33-64	6
65-128	7
129-256	8
257-512	9
513-1024	10
1025-2048	11
2049-4096	12
4097-8192	13

Magnetic Tape Characteristics for Sort 3X Timing Purposes

Description	Tape Unit	163	164	603LD	603HD	606LD	606HD	1607
CPR	Characters per reel	5.52M	5.52M	5.52M	15.3M	5.52M	15.3M	5.52M
IBS	Characters for inter block space	150	150	150	417	150	417	150
TR	Character transfer rate (KC)	30	15	15	41.5	30	83	30
RS	Read Start (MS)	5	7	7	7	5	5	5
WS	Write Start (MS)	18	27	7	7	5	5	7
SD	Stop Delay (MS)	10	10	0	0	0	0	0
RW	Rewind time for full reel (minutes)	3.33	2	1.33	1.33	1.33	1.33	3.33

Note: 5.52 M = 5,520,000

15.3 M = 15,300,000

14. Equipment Configuration

Minimum SWAP 160 or 160-A computer and four magnetic tape units. (163, 164, 1607 or 606)

METHOD

The sort process used in Sort 3X is performed in three steps. The first called Phase 1 is an internal sort which reads individual records, determines the sort control fileds, and sorts internally to form sorted groups of records. Phase 2 is a two way merge of the individual sorted groups which were formed in Phase 1. Phase 3 performs the final two way merge to form the sorted file and also writes out the individual records in their original form. It should be noted that although the input and output of the Sort 34 program are individual records, the intermediate tapes formed between the various phases of the program are blocked up to form records of 1,000 characters for maximum magnetic tape efficiency.

Sorting Technique

The sorting technique used in the Sort 3X program consists of reading input records from the input file until the total of records plus key is a total of 3,000 characters. These records are then arranged in sequence and written on alternate tapes in blocks of 1,000 characters of record plus key. After the input file has been read in, phase 2 of the sort program merges these sorted sequences into longer sequences. When there is one merge left, the sort 34 program automatically goes into phase 3 which produces one long sequence which is the sorted file.

One tape is used for input, and two units are used for output during the initial sorting process. In the merging process, the input units become output units, and vice versa.

Sort 3X accomplishes the sorting operation in three steps - Phase 1, phase 2, phase 3.

Phase 1

1. Phase 1 determines the total size of the sort key filed and reserves space prior to the data for the key field.
2. Each record is read into the area immediately following its reserved key field, the size of the record is determined from the read in command.
3. The key is extracted from the record, recoded to a sort sequence code, and placed in front of the record in its key field. The key information is extracted from up to ten fields in the record.
4. Successive records are read in until the storage occupied exceed 3,000 characters for key and records.
5. The last record read in is backspaced to be accepted as the first record of the next block.

6. The information is sorted internally to form a sequence of not over 3,000 characters in final sort order.
7. The information is written on the output tape in blocks of not over 1,000 characters. The blocking factor is determined solely by the number of records plus key which can be contained in less then 1,000 characters.
8. Output is to the alternate output tape when there is a break in sort sequence between the last record output from the previous block and the birst record output of the current block.
9. A sequence count is kept so there will be an automatic switch to phase 3 if the records were in sort on input, or were brought into sort by the internal sort process.

Phase 2

1. Phase 2 merges the sequences written during phase 1 using as many merge passes as are required to arrive at two sorted files, one one each of the two output tapes. The files are still blocked to a maximum of 1,000 characters per block. The blocking information is contained on the magnetic tape blocks.
2. On reaching two sorted files, phase two goes to phase three.

Phase 3

1. The two sorted files are merged to form one sorted output file.
2. At output, the key is stripped from the record, and the original record is written on the output tape.
3. If at any time during the sort process in phase 1, 2 or 3, a record was encountered which was not recoverable during the read process, the record is flagged as possibly being wrong. The bad records are written on an alternate output tape as well as being written on the final output tape. This allows for the operator to peruse the possible errors and take corrective action.

Allowable Input Record Configurations

Sort 3X accommodates variable length input records which must appear on tape singly as one record per block of tape. The maximum size of an input record must be such that the number of characters in the record plus the number of characters in the key do not exceed 984 characters.

Maximum File Length

The input file to be processed by Sort 3X must be no longer than the number of records that can be contained on a single tape reel when in the blocked form for the merge passes. This number depends on the record length, key length, and whether the processing is being performed on a high density or low density magnetic tape mode. The following table shows the maximum number of records which can be processed as one job.

Table 1

Key and record size in characters	Maximum number of records	
	Low density	High density
10	450,000	1,000,000
20	225,000	500,000
30	150,000	333,300
40	111,000	250,000
50	90,000	200,000
60	75,000	166,700
70	64,200	141,500
80	56,500	125,000
90	50,000	111,100
100	45,000	100,000
120	37,500	83,000
140	32,100	71,500
160	28,000	62,500
180	25,000	55,500
200	22,500	50,000
300	15,000	33,300
400	11,100	25,000
500	9,000	20,000
600	7,500	16,700
700	6,400	14,100
800	5,600	12,500
900	5,000	11,100

EXAMPLE

Compute the maximum file size for records 50 characters long using a sort key of six characters. To total record plus key size is 56 characters. Assume high density mode of processing.

From the table we find that for 60 characters we have a file size of 166,700 records and for 50 characters we have a file size of 200,000 records thus we can use a file of approximately 180,000 records.

Sort Key Fields

From one to ten fields of each input record can be specified to control sequencing of the file. These fields can be located anywhere within the record, provided they are in the the same place in each record. The fields can be of any length.

The locations of each control field is specified by the contents of locations 0100, 0101 for the first field, 0102, 0103 for the next lower field, etc. in the core storage of the 160 or 160-A computer. The values of these locations can be very conveniently set up by the operator by following the operating instructions.

There is no penalty for using any number of fields or the location of the fields in the record. The control fields can contain any character. The standard collating sequence is included in the program, but can be changed by changing the contents of the location 0200 to 0277 in the Sort 3X program.

Unreadable Input Records

Input tape blocks, during any phase, are flagged for special treatment on output. An unreadable record is one that causes redundancy error indications after eight attempts to re-read the input record. A read error does not cause the sort program to stop. In the phase 3 (output phase) the unreadable records are put out with the regular sorted file and they are also listed on the error tape.

INTERMEDIATE TAPE FORMATS

The intermediate tapes produced by sort 3X are written in binary and are provided with a header record and a trailer record. The tape records in between also are blocked records. All records are a multiple of two characters and will be shown in a two character representation as seen in memory. This is also the form of the punched paper tape output of an error from the 3316 stop.

The formats are as follows:

Header Record

7772	Header code identifying this record as a header
00xx	Tape account number. Starts at 1 and is increased by 1 for each tape written.
yyyy	Key size. Size of sort key in words for rerun of the program.

Information Record

7775	Beginning of information block code
xxxx	Number of records in block
7776	Beginning of record, may be 7774 if record came from parity error block
yyyy	Number of computer words to reach next record
kkkk	
.	Key information, coded in sort order code
.	
.	
kkkk	
rrrr	
.	Original record
.	
.	
rrrr	
7776	Beginning of next record
yyyy	
kkkk	
.	
.	
.	
rrrr	
.	
.	
.	
7773	End of block code (placed where next beginning of record code might be)

Trailer Record

7771	Beginning of trailer code
0000	
7773	
0000	
bbbb	Count of number of information blocks on the tape
bbbb	
rrrr	
rrrr	Count of number of information records on the tape
rrrr	
	The counts in the trailer record are carried in BCK notation.

SAMPLE TAPE

A sample intermediate tape containing the records 2abc, 3abc 1abc which are being sorted on column 4 and 1 would appear on an intermediate tape in the following form, assuming that the block size was two records.

Header

7772	
00xx	Tape account number
0001	Sort key is one word or two characters

Information Block 1

7775	
0002	Two records on this block
7776	Beginning of record flag
0005	This record is five words long, including key and flag
1503	Sort key "c2" in sort order code
0261	"2a" in magnetic tape code
6263	"bc" in magnetic tape code
7776	Beginning of second record flag
0005	
1504	
0361	
6263	
7773	End of block flag

Information Block 2

7775	
0001	
7776	
0005	
1502	
0161	
6263	
7773	End of block flag

Trailer Record

7771		
0000		
7773		
0000		
0002	Low order	3 decimal digits of block count given in octal
0000	High order	3 decimal digits of block count given in octal
0003	Low order	3 decimal digits of record count given in octal
0000	Middle order	3 decimal digits of record count given in octal
0000	High order	3 decimal digits of record count given in octal

The beginning of information record code 7776 may be changed to 7774 by sort 3X if the record is in doubt. The value 7774 is used as a tape as well as the normal record tape.

Appendix I

Sort 3X Halt Table

P	A	Reason and Corrective Action
0134	1152	Error tape 2 rewound on 162, 163, or 164 tape system. List tape number 2 and take corrective action on sorted file.
0141	1154	Error tape 4 rewound on 162, 163, or 164 tape system. List tape number 4 and take corrective action on sorted file.
9426 n/2		Source input record is less than 16 characters. The contents of A gives one half of the number of characters read in the input record. If this record is legal, run to accept. To delete the record from the sort, set A to 4000 or greater and run. This check is included to catch possible noise records from the input change location 0423 to 070N where N is one half the size of the shortest expected record and start at P = 0430 to accept the record which caused the halt prior to correcting the program.
0701	0001	Sort 3X completed. Output on tape number 1. To rewind error listing tape, see step 1.e.9 of operating procedure.
0711	0003	Sort 3X completed. Output on tape number 3. To rewind error listing tape, see step 1.3.9,of operating procedure.
0762	abcd	This is optional stop 7701 which occurs between merge passes. For 160, to eliminate this stop change location 0762 to 0300. Start at P = 0761. The digits in the A register indicate the input tapes as a, b, and the output tape as c, d which were used in the last merge pass. This stop may be used as a rerun point in a long sort. Remove tapes a and b and replace with scratch tapes. To continue operation, do not change the contents of A, and run. For the 160-A, to eliminate this stop, return optional stop switch 1 to the neutral position.
1036	xxxx	Input tape A header halt. Checks on the header of the intermediate input tape header format failed for the following reasons: (1) Unrecovered parity error in reading the input header. (2) Account number on the header did not compare with the internal account number. (3) No header record on the tape. Replace tape A as indicated by octal digit Axxxx in location 0037. Manual rewind load 11 tapes. Start with

P = 0761. Program will stop with P = 0762 if optional stop 1 is on. The program will stop with P = 1036. Run, program will stop with P = 1067. Run.

This recovery procedure assumes that the scratch tape was bad and performs the previous merge pass. The two stops at 1036 and 1067 will be caused by account number errors which should be ignored.

1067 xxxx Input tape B header halt.

Checks on the header of the intermediate input tape header format failed for same reasons as 1036 halt.

Replace tape B as indicated by octal digit xBxx in loca- 0037. Follow rest of the recovery procedure given for the 1036 halt.

1077 xxxx Key size specified on intermediate tape A and B do not agree.

If this stop occurs during a normal sort, manual rewind load all tapes and start with P = 0761. Program will stop with P = 0762 if optional stop 1 is on. The program will stop with P = 1036. Run, program will stop with P = 1067. Run.

If this stop occurs during the resumption of an interrupted sort, obtain the correct pair of input tapes and restart the resumption procedure. The tapes presently on drives 1 and 2 are not the correct pair.

1104 xxxx Key size specified on intermediate tapes and in the memory of computer do not agree.

This stop should occur during the resumption of an interrupted sort. Run to get correct key size in memory from tape. This stop should not occur otherwise.

1611 1611 Intermediate output tape C at physical end of tape.

Place longer scratch tape on unit C indicated by octal digit xxCx in location 0037. Manual rewind load all tapes. Start with P = 0777.

1621 xxxx Attempt to write merge output on error tape during third phase, or output section of sort.

This error may occur as a result of a recovery procedure. Manual rewind load all tapes. Start with P = 0777.

1700 1700 Intermediate output tape D at physical end of tape.

Place longer scratch tape on unit D indicated by octal digit xxxD in location 0037. Manual rewind load all tapes. Start with P = 0777.

2004 xxxx Intermediate tape A record count error.

Number of records moved from A input area does not compare with the number indicated on the input tape for this block. xxxx = number of records in the block minus the number of records moved out.

This stop usually is an indication of machine trouble, however one attempt may be made to get around the error by manual rewind load all tapes and start with P = 0777.

2103	xxxx	Intermediate tape A file accounting error. Number of records and blocks processed from intermediate tape A does not check with the Tape A trailer account records. Manually rewind load all tapes and start with P = 0777.
2137	xxxx	Format error. The intermediate record read in from tape A does not conform to the format required for sort 3X. Manual rewind load all tapes. Start with P = 0777.
2163	xxxx	Intermediate tape B record count error. See 2004 stop for cause and action.
2262	xxxx	Intermediate tape B file accounting error. See 2103 stop for cause and action.
2316	xxxx	Format error. The intermediate record fed in from tape B does not conform to the format required for sort 3X. Manual rewind load all tapes. Start with P = 0777.
2337	xxxx	
2352	xxxx	
2363	xxxx	
2402	xxxx	
2606	xxxx	
2611	xxxx	All above stops may occur only during the first or input pass and have to do with internal format checks Manual rewind load all tapes. Start with P = 0000.
2764	xxxx	Intermediate pass output tape is at physical end of tape during input pass. If contents of 0031 = 0001, output is tape D. If contents of 0031 = 0000, output is tape C. Place a longer scratch tape on output tape c or d as indicated by the contents of location 0037 = xxCD. Manual rewind load all tapes. Start P = 0000.
3007	xxxx	Internal format error during output pass. Manual rewind load all tapes. Start P = 0770.
3050	0040	Physical end of tape normal output tape C during output pass. Place longer tape on output tape C as indicated by the contents of location 0037 = xxCx. Manual rewind load all tapes. Start P = 0770.
3076	0040	Physical end of tape, error output tape D during output pass. Place longer tape on output tape D as indicated by contents of location 0037 = xxxD. Manual rewind load all tapes. Start P = 0770.
3312	xxxx	Halt before re-read of bad input record. A contains the address of the location in Sort 3X where re-read of the input record will start. This stop always follows a 3316 stop in which the decision is to reread the input tape. Run to continue.

3316 7777 Format error on reading intermediate input tape.
The block read in from the intermediate tape does not conform to the format required by sort 3X. This stop is caused by either a noise record left on the tape in passing a bad spot, or by a malfunction of the tape system.

Recommended recovery procedure is:

1) Re-read the bad record. Move run-step switch to neutral and then to run. Computer will arrive at 3312 stop. Run again to reread the bad tape. If the 3316 stop occurs again, it is probably a noise record, so take step 2.

2) Eliminate the record from the operation. Turn on paper tape punch. Clear A register and run. The bad record will be punched out on paper tape in the magnetic tape code. Feed out the paper tape and keep the bad record for possible diagnostics by the maintenance representative.

3) If this stop persists on alternate passes, it is possible that one of the scratch tapes has a bad area on it. Note the tape unit which is used during the tape re-read process, set optional stop 1 and replace the tape at the next between merge stop. Caution. Replace the bad tape at the between merge stop immediately following the 3316 stop and recovery procedure.

4) If a 2103 or 2262 stop occurs after the 3316 stop, the record removed from the sort is one of the records in the file. It is recommended that the sort be performed again.

3720 xxxx Error tape rewound on 1607 or 1615 system.
List error tape and take corrective action on sorted file.

I1.
IDENTIFICATION

Title: Proper Fraction for Calculating Percentages - Ident. I1.00
Programmer: L. Goodreau, May 1961

PURPOSE

Division of $\frac{x}{y}$ where x is assumed to be $<$ y. The answer will be in the form .xxxx, adjusted by rounding, in true binary.

USAGE

1. Operational Procedure: Program tape can be loaded anywhere in memory.

2. Arguments

   a. Dividend in location 71
   b. Divisor in location 70
   c. Exit to main program in 77
   d. Result in location 73

      1) If the divisor = 0, result will be 0.
      2) If x < y, the result will be in the form .xxxx, rounded.
      3) Space required: 196_{10} locations, including locations 70 - 77.

METHOD

D = dividend which is $\leq 4095_{10}$

d = divisor which is $\leq 4095_{10}$

D BCK x 10^3

d BCK

d BCK x 10^1

d BCK x 10^2

I1.

IDENTIFICATION

Title: Single Precision Divide - Ident. I1.01
Programmer: Sanford Elkin, February 1961

PURPOSE

This subroutine will divide a positive 23-bit fraction by a positive 11-bit fraction, giving a rounded 11-bit fractional quotient.

USAGE

1. Operational Procedure

   The dividend (a and b) must be placed in locations 60 and 61 respectively, and the divisor (x) in 62. The contents of a must be less than the contents of x, and both must be positive. b contains 12 low order magnitude bits, which may be all zeroes. The routine is entered at the symbolic address DVDSBR, and the return address must be in cell 7. The quotient (y) will be in cell 63.

3. Space Required: $27_{10} = 33_8$ locations 7 and 60-63.

11. Timing: 2.0 milliseconds

12. Cautions to User: The dividend is destroyed by the subroutine. If the true quotient equals or exceeds $1 - 2^{-12}$, the octal value 4000 will be in cell 63.

13. Accuracy: 1 bits, with answer being rounded.

14. Equipment Configuration: Minimum SWAP 160 Computer.

MATHEMATICAL METHOD

Repeated subtractions are used.

I1.

IDENTIFICATION

Title: 9-bit Quick Multiply - Ident. I1.02
Programmer: Sandord Elkin, February 1961

PURPOSE

This subroutine will multiply two signed 11-bit numbers together in about 600 microseconds, giving a signed answer accurate to approximately 10 bits.

USAGE

1. Operational Procedure

   The two numbers and their product are interpreted as signed fractions with magnitude less than unity. The multiplicand must be in cell 10 and the multiplier in cell 11. Cells 70-72 are used for temporary storage, and the product will be placed in the A register. The routine is entered at the symbolic address MPY9B, and cell 2 must contain the return address.

3. Space Required: 181_{10} or 265_8 locations, plus locations 10, 11, and 70-72.

11. Timing: Average - 595 microseconds, maximum - 660 microseconds

13. Accuracy: Average - 1×2^{-11}, worst case - 3×2^{-11}

14. Equipment configuration: Minimum SWAP 160 Computer

MATHEMATICAL METHOD

2^{-10} is added to the absolute value of each factor, and the two low-order bits of each are truncated. If either resulting number is 0 or 1, 0 or the other factor becomes the answer. Each of the three octal digits of the resultant multiplier is examined, and the partial product of each with the resultant multiplicand is obtained. The bits in each partial product which are less significant than 2^{-12} are truncated, and the sum of the partial products is truncated to 2^{-11}. The sign of the product is then obtained, and the answer is placed in the A register.

I1.

IDENTIFICATION

Title: Integer Divide - Ident. I1.03
Programmer: Sandord Elkin, March 1961

PURPOSE

This subroutine will divide a positive 23-bit integer by a positive 11-bit integer, giving a 12-bit quotient with an 11-bit remainder.

USAGE

1. Operational Procedure

   The 23 bit integer must be placed in a (70) and b (71), the divisor in x (72), and the return address in exit (7). x must be greater than a (a being the more significant half). The routine must be entered at DVDINT. The quotient will be in y (73) and the remainder in a (70).

3. Space Required: $22_{10} = 26_8$ locations.

4. Temporary Storage: Octal locations 7 and 70-74.

11. Timing: 2.0 milliseconds.

14. Equipment Configuration: Minimum SWAP 160 computer

MATHEMATICAL METHOD

Repeated subtraction are used.

I1.

IDENTIFICATION

Title: Fractional Arithmetic - 22 bit - Ident. I1.04
Programmer: S. Palais and J. Seewerker, July 1961

PURPOSE

This is a relocatable package, designed to add, subtract, multiply or divide 22-bit binary fractions (positive or negative).

USAGE

1. Operational Procedure

a. Calling Sequence

The desired subroutine may be entered by either a JPI or a JFI 01 as in the example below.

	Location	Subroutine Address
ADD	0002	105
SUBTRACT	0003	100
MULTIPLY	0004	220
DIVIDE	0005	401
RETURN ADDRESS	0007	---

Examples:

```
LDF    03                    LDF   04
STD    07                    STD   07
JPI    (02, 03, 04 or 05)    JFI   01
       (RETURN ADDRESS)            (SUBROUTINE ADDRESS)
                                   (RETURN ADDRESS)
```

The augend, multiplicand, minuend or dividend is placed in cells 70 and 71 and the addend, subtrahend, multiplier or divisor in cells 72 and 73. The result appears in 70 and 71, with the contents of 72 and 73 unchanged. Thus, a sequence of operations can be performed, changing only the contents of 72 and 73. An overflow in addition or subtraction is indicated by a 1 in cell 74.

b. Operating Instructions

If the 22-bit Fractional Arithmetic non-relocatable program tape is

loaded at 00, it may be run as is. Since the executable part of the routine can be relocated anywhere above 100 one can load the relocatable tape anywhere and then add the new load address minus 100 to the entrance addresses given above and enter the resulting sums in registers 0002 thru 0005. This is the only change necessary to relocate the program.

3. Space Required

a. Low Core

Besides locations 0002 thru 0005 and 0007 specified above, cells 70 thru 77 are used as temporary storage.

b. High Core

The executable subroutines start at location 100 and run to location 573. The following is a breakdown for the various subroutines:

Subroutines	Entrance Address	Space Required Octal	Decimal
22 ADD	0105	113	75
22 SUB	0100	120	80
22 MULT	0220	161	113
22 DIV	0401	173	123

with a total length of $316_{10} = 474_8$.

8. Error Return

If the divisor is less than or equal to the dividend, an error return is signaled by a non-zero accumulator upon return from the subroutine (a normal return from a subroutine will have a zero in the accumulator). The input to registers 70 thru 73 is restored and if the dividend equals the divisor, location 74 will contain 4000.

10. Input and Output Format

a. Positive 22-bit Fractions

The most significant 11 bits of the fraction are placed in the "high order" word and the least significant 11 bits in the "low order" word with the left-most bit of both words being zero; e.g., in the division routine, cells 70 and 71 would contain the high and low order parts of the dividend respectively and cells 72 and 73 the high and low order parts of the divisor.

b. Negative Fractions

The add and subtract routines do 24 bit complementary arithmetic. Thus, a negative 22 bit fraction is broken up into the high and low order parts as above and then one's complement of all 24 bits is taken and entered into the high and low order words (which the computer will then regard as negative since the left-most bit of each word will contain a one bit.) Thus, if the result appearing in 70 and 71 is negative, the magnitude of the number may be found by complementing both the high and low order words. The high and low order parts of an input number must agree in sign, otherwise an incorrect answer will be given.

11. Timing

Subroutine	Execution Time
22 ADD	225 microseconds
22 SUB	300 microseconds
22 MULT	5 - 10 milliseconds
22 DIV	10 miliseconds

12. Cautions

Since cell 74 (the overflow indicator) is set equal to zero at the beginning of each addition, subtraction or division, the user should check for overflow at the end of each of these operations if he intends to do a sequence of computations. In using any other subroutines related to 22-bit arithmetic, one should be careful to distinguish whether they are applicable to 22-bit fractional arithmetic.

13. Accuracy

All subroutines result in 22-bit positive fractions or 24-bit one's complement negative fractions. In multiplication, the product is rounded to 22-bits. In division, the quotient is truncated to 22-bits.

14. Equipment Configuration: Minimum SWAP 160 or 160-A Computer

I1.

IDENTIFICATION

Title: Nine-Bit Quick Sine - Ident I1.05
Programmer: S. Elkin, August 1961, Control Data Corporation

PURPOSE

This subroutine will obtain the sine of any angle whose magnitude is less than 90°, in about 3 milliseconds, with the answer accurate to about 9 bits.

USAGE:

1. Operational Procedure

The angle x must be put in the A register before entering the routine, and the answer sin x will be put in the A-register. The format of x must be that of a signed binary fraction, i.e. $4000_8 \leq x \leq 3777_8$, and it is interpreted as a fraction of 90°. For example, $x = 2000_8$ means $x = 45^{\circ}$, and $x = 4000_8$ means $x \simeq -89.956^{\circ}$. Sin x should also be interpreted as a signed binary fraction. The routine is entered at the symbolic address SINFX, and XTSIN (location 3) must contain the return address. Finally, ADRMPY (location 4) must contain the address of the 9-bit Quick Multiply.

3. Space Required

For the Sine routine alone, $38_{10} = 46_8$ locations, plus location 3 for the exit address. However, the 9-bit Quick Multiply requires $181_{10} = 265_8$ locations plus 5 locations in low core.

11. Timing

Approximately 2.85 ms average, 3.1 ms maximum.

13. Accuracy

Average error from 0° to 67.5° is less than .001, from 67.5° to 80° is less than .002, and between 80° and 90° is less than .004.

14. Equipment Configuration: Minimum SWAP 160 or 160-A computer.

16. Reference: "Some Basic 1604 Mathematical Subroutines," (CDC Publication 061).

MATHEMATICAL METHOD

The Tchebychev equation

$$\sin \frac{\pi}{2} x = x(C_3 - x^2(C_2 - x^2(C_1)))$$

is used, where $C_3 \simeq 1.5703$, $C_2 \simeq .6419$, $C_1 \simeq .0715$, and $-1 < x < 1$.

The machine implementation of the equation is

$$\sin x = 2x(3110_8 - x^2(1222_8 - x^2(0111_8))),$$

where x is a binary fraction of $\frac{\pi}{2}$.

I1.
IDENTIFICATION

Title: Matrix Inversion - Ident. I1.06
Programmer: D. M. Lytle, September 1962, CODA

PURPOSE

This subroutine is written in 160 FORTRAN (160 FORTRAN-A) for finding the inverse of a square matrix.

USAGE

1. Calling Sequence: The following two statements must be in the main program.

```
DIMENSION  A(n,n),L(n),M(n)
N = n
```

where n = size of square matrix.

The inverse matrix of A is stored back into A. Communication between the subroutine and main program is accomplished by the NØN LØCAL statement in the subroutine.

3. Space Requirement: The subroutine requires 747_{-8} (487_{10}) locations. A 17x17 matrix could be inverted in the 160 and 33x33 for 160-A.

11. Timing:

Size	
3x3	3 seconds
10x10	53 seconds
20x20	6½ minutes
30x30	20(estimated minutes)

16. Reference: Lanczoc, C., Applied Analysis, Prentice Hall, Inc., 1956, 118-122.

METHOD

The Gaussian Elimination Method is used for the inversion.

```
C     PROGRAM FOR FINDING THE INVERSE OF A NXN MATRIX
256    FORMAT(  3E20.5      )
      DIMENSION A(3,3),L(3),M(3)
      SUBROUTINE INVER
      NONLOCAL A,L,M,N
C     SEARCH FOR LARGEST ELEMENT
      D=1.0
      DO80 K=1,N
      L(K)=K
      M(K)=K
      BIGA=A(K,K)
      DO20 I=K,N
      DO20 J=K,N
      IF(ABSF(BIGA)-ABSF(A(I,J))  10,20,20
   10 BIGA=A(I,J)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF(L(K)-K) 35,35,25
   25 DO30 I=1,N
      HOLD=-A(K,I)
      A(K,I)=A(J,I)
   30 A(J,I)=HOLD
C     INTERCHANGE COLUMNS
   35 I=M(K)
      IF(M(K)-K) 45,45,37
   37 DO40 J=1,N
      HOLD=-A(J,K)
      A(J,K)=A(J,I)
   40 A(J,I)=HOLD
   45 DO55 I=1,N
   46 IF(I-K)50,55,50
   50 A(I,K)=A(I,K)/(-A(K,K))
   55 CONTINUE
C     REDUCE MATRIX
      DO65 I=1,N
      DO65 J=1,N
   56 IF(I-K) 57,65,57
   57 IF(J-K) 60,65,60
   60 A(I,J)=A(I,K)+A(K,J)+A(I,J)
   65 CONTINUE
      DO75 J=1,N
   68 IF(J-K)70,75,70
   70 A(K,J)=A(K,J)/A(K,K)
   75 CONTINUE
      D=D+A(K,K)
      A(K,K)=1.0/A(K,K)
   80 CONTINUE
C     FINAL ROW AND COLUMN INTERCHANGE
```

must be defined by main program

subroutine

```
      K=N
  100 K=(K-1)
      IF(K) 150,150,103
  103 I=L(K)
      IF(I-K) 120,120,105
  105 DO110 J=1,N
      HOLD=A(J,K)
      A(J,K)=-A(J,I)
  110 A(J,I)=HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 DO130 I=1,N
      HOLD=A(K,I)
      A(K,I)=-A(J,I)
  130 A(J,I)=HOLD
      GO TO 100
  150 CONTINUE
      RETURN
      END
```

I1.
IDENTIFICATION

Title: Program Structure - Ident. AI1.07
Programmer: Elizabeth Phillips, October 1962, MSU - Computer Laboratory

PURPOSE

This program determines all those permutations π which leave a structure (S,R) invariant, and all those permutations ρ which leave (S,R) self-dual.

A structure (S,R) is a finite set S (which may as well be the first n positive integer) together with a relation R on the set, which is symmetric and reflexive, though not necessarily transitive. The permutation π leaves (S,R) invariant if aRb if and only if $\pi(a)R\pi(b)$; ρ leaves (S,R) self-dual if aRb if and only if $\rho(a)$ $\rho(b)$, in both cases for all a,b in S except for a = b.

A matrix C(S,R) is associated with (S,R) where C(S,R) (a_{ij}) and a_{ij} =1 if iRj; a_{ij} = 0 if i = j. Then π leaves (S,R) invariant if and only if $a_{ij} = a_{\pi(i)\pi(j)}$; and ρ leaves (S,R) self-dual if and only if $a_{ij} = \bar{a}_{\rho(i)\rho(j)}$ of all i and j in S except i = j. Here $\bar{a}_{\rho(i)\rho(j)}$ means the Boolean Complement.

This program accepts a matrix C(S,R) and produces all permutations which leave (S,R) invariant and all those (if there are any) which leave (S,R) self-dual.

USAGE

1. Operation Procedure:

   Master clear, clear both memory banks. Load Fortran Interpreter (AC1.00-3D) according to operating instructions for 160 Fortran-A. Input object code. At stop, position data tape in reader and run.

7. Error stops: Those listed for Fortran.

10. Input and Output Formats:

   (i) Carriage return. A two digit integer equal to the order of the matrix to be permuted.

   (ii) Carriage return. A two digit integer, either "00" or "01." Use "01" if intermediate output is desired after every 1000th permutation for stopping and restarting purposes. For cases of n 8, this would not be necessary.

a. Input (cont.)

(iii) Carriage return. A one digit integer for each element of the matrix. These are read in by columns, each element preceded by a carriage return. There will be n^2 of these.

(iv) Carriage return. A two digit integer for each member of the first permutation to be tried. Each integer is preceded by a carriage return. There will be n of these.

These will be of the form $n_1, n_2, n_3, \ldots, n_q$ which means the permutation

$$\begin{matrix} 1 & 2 & 3 & q \\ n_1 & n_2 & n_3 & n_q \end{matrix} .$$

(v) If (ii) is "00", this completes the data. If (ii) is "01", additional information needs to be input--to allow for starting in the middle of the run. This consists of a carriage return and a two digit integer equal to "00" only if it is desired to start at the beginning of the problem; "01", to start in the middle. Then for the initial input, 2(n-1) "00"'s are punched, each "00" being preceded by a carriage return. On successive runs, these values will have been output (thus telling you how far the problem got and from where to continue next).

b. Output

The headings "Invariant Permutation" or Self-Dual Premutation" are punched out followed by the order of the matrix (n) and the n components to be interpreted as follows:

$$\begin{array}{lcl} n & : & 03 \\ x_1 & : & 03 \\ x_2 & : & 01 \\ x_3 & : & 02 \end{array}$$

this means the matrix is of order 3, the permutation is 03 01, 01 02, 02 03, or

$$\begin{matrix} 1 & 2 & 3 \\ 3 & 1 & 2 \end{matrix} .$$

If (ii) on the input tape was "01" then after every 1000th permutation (for n=8, this would be 40 times) there will be 2(n-1) two digit integers output (leading zeroes are supressed). This is the information that is put back in in step (v) above, preceded by "01", if restarting at this point is desired.

11. Timing: For n=5, the time is 2.5 minutes; for n=6, 18.75 minutes; for n=8, approximately 25 hours (thus the need for restart procedures).

14. Equipment Configuration: minimum SWAP 160-A computer.

15. SWAP Routines Used; 160 Fortran A.

16. References: H. F. Trotter, "Algorithm 115, PERM", Comm. ACM, August, 1962, pp.434-5.

METHOD OR ALGORITHM:

The method used to obtain the n! permutations of the set of n integers is one developed by H. F. Trotter. It changes the order of the given set once each loop around, so that after n! loops, all n! permutations have been generated. For each permutation generated, the original matrix, M (which is input), is permuted, first by rows, then columns. This new matrix is then compared with the original matrix, element by element. If the new matrix is exactly like the original, the output represents a permutation which leaves the matrix invariant. If the new matrix is the complement of the original excluding diagonal elements, the output is printed out as self-dual. In comparing the two matrices, if at any time an element is discovered which contradicts either of these restrictions, the comparison is abandoned and a new permutation is generated.

```
C                 PROGRAM STRUCTURE
 0140             format (21hinvariant;permutation)
 0141             format (21hself;dual;permutation)
 0100             format (i2)
 0101             format (i1)
 0102             format (/)
 0103             dimension k(24), is(12), js(12), m(12,12), mp(12,12), mpp(12,12)
 0104             read 100, n,lstop
 0105             n2 = 2'n
 0106             n3 = n+1
 0107             read 101,((M(i,j),i=1,n),j=1,n)
 0108             read 100, (K(i),i=n3,n2)
 0109             do 110 i=1,n
 0110             k(i) = i
 0111             LFIRST = 0
                  if (LSTOP) 31,4,31
 0031             read 100, lfirst, (IS(I), JS(I),i=2,n)
 0032             kount =0
C                 BEGIN PERMUTE N NUMBERS
 0004             nt = n
 0005             if (LFIRST) 10,6,10
 0006             do 8 i=2,n
 0007             is(i) = 0
 0008             js(i) = 1
 0009             LFIRST = 1
 0010             m 1= 0
 0011             L = IS(NT) +JS(NT)
                  IS(NT) = L
 0012             if (L-NT) 15,13,15
 0013             JS(NT) = -1
 0014             go to 18
 0015             if (L) 23,16,23
 0016             JS(NT) = 1
 0017             M 1= M1+1
 0018             if (NT-2) 21,21,19
 0019             NT = NT-1
 0020             go to 11
 0021             L = 1
 0022             LFIRST = 0
 0023             L = L+M1
 0024             n1 = n+L
 0025             kt = K(N1)
 0026             K(N1) = K(N1+1)
 0027             K(N1+1) = KT
 0028             if (LSTOP) 33,29,33
 0033             kount = kount +1
 0034             if (KOUNT- 1000) 29,35,29
 0035             kount =0
 0036             punch 100, (IS(I), JS(I), i=2,n)
```

```
                    punch 102
c                   BEGIN PERMUTE MATRIX
 0029               do 116 i=1,n
                    in = I+N
 0112               nl = K(I)
 0113               L = K(IN)
 0114               do 115 j= 1,n
 0115               mp(nl,j) = M(L,j)
 0116               continue
 0117               do 123 i=1,n
                    in = I+N
 0119               nl = K(I)
 0120               L = K(IN)
 0121               do 122 j =1,n
 0122               mpp(j,nl) = MP(j,L)
 0123               continue
C                   COMPARE MATRICES FOR INVARIANCE+ SELF-DUALITY
 0125               do 128 i=1,n
 0126               do 128 j=1,n
 0127               if (M(i,j) - MPP(i,j)) 180,128,180
 0128               continue
                    punch 140
                    go to 129
 0180               do 134 i=1,n
                    do 134 j=1,n
 0131               if (I-J) 132,133,132
 0132               if (M(i,j) - MPP(i,j)) 134,170,134
 0133               if (M(i,j) - MPP(i,j)) 170,134,170
 0134               continue
                    punch 141
 0129               punch 100, n, (K(I), i=n3,n2)
 0130               punch 102
 0170               if (LFIRST) 4,144,4
 0144               pause 0001
                    go to 104
                    end
                    end
```

J1.
IDENTIFICATION

Title: LINEAR PROGRAM I (SIMPLEX METHOD) - Ident J1.00.
Programmer: John F. Quinlan, August 1962, Control Data Corporation

PURPOSE

This program is written in FORTRAN for the Control Data 160 or 160-A computer. The Simplex Method is the computational technique used and the input data must inply a first feasible solution. This condition is not limiting since any LP problems can be arranged to meet it by the proper manipulation of input parameters. Examples of the manipulation are given in the sample problem.

USAGE

1. Operational Procedures

   Using technique for FORTRAN, the first set of restraints and minimizing coefficients are read in. After punching out answer, computer will stop (A=1111). The next set of coefficients can be placed in reader and run. Computer will then repeat -, punch answers, and stop (A=1111).

3. Space Requirements

   The size of the LP problems handled by the program are given by the equation

$$2(M+N) + (M+1)(N+1) \quad G$$

   where G = 245 for 160 computer
   G = 1117 for 160-A computer
   G = (estimated 6 banks) 6237 for 160-A FORTRAN
   M = number of restraining equations
   N = number of non-slack variables

   Slack variables either natural or artificial, are not included in assessing N.

Sample Problem I

Minimize

$$x_1 + x_2 + x_3$$

Subject to

$$(1)\ -x_1 + x_2 - x_3 \leq -2$$
$$(2)\ -x_1 - x_2 + x_3 \leq 1$$

Since restraint (1) has a right hand side less than zero, multiply it through by -1 giving

$$(1)\ x_1 - x_2 + x_3 \geq 2$$

Since the relationship is now one of $\geq$, another non-slack variable x_4 is introduced giving

$$(1)\ x_1 - x_2 + x_3 - x_4 = 2$$

Because an equality exists, it is necessary to associate an artificial slack with this restraint. Thus the inputs will be taken from the following system:

Minimize

$$10a_1^* + x_1 + x_2 + x_3$$

Subject to

$$(1)\ a_1 + x_1 - x_2 + x_3 - x_4 = 2$$
$$(2)\ y_2 - x_1 - x_2 + x_3 = 1$$

* x_1 is placed here as the artificial slack is also to be minimized. 10 is arbitrarily chosen, 100 would do just as well.

Sample Problem I (continued)

Where x_1 is an artificial slack and

x_2 is a natural slack

The identifiers on output will be as follows:

Variable	Identifier
a_1	1
y_2	2
x_1	3
x_2	4
x_3	5
x_4	6

In this problem M = 2, N = 4

The input and output for this problem are shown below

input		output
0204/	M, N	
10.0e0/		3
0.0e0/		.20000000e 01
1.0e0/	Z	2
1.0e0/		.30000000e 01
1.0e0/		.20000000e 01
0.0e0/		
1.0e0/		
-1.0e0/		
1.0e0/	(1)	
-1.0e0/		
2.0e0/		
-1.0e0/		
-1.0e0/		
1.0e0/	(2)	
0.0e0/		
1.0e0/		

NOTE:

Coefficients for a_1 and y_2 are not included here

Execution Time -- 9 seconds

Sample Problem II*

In this problem, M = 5, N = 4.

Maximize

$x_1 + 2x_2$

Subject to

(1) $-x_1 + 3x_2 \leq 10$ with all $x_i \geq 0$
(2) $x_1 + x_2 \leq 6$
(3) $x_1 - x_2 \leq 2$
(4) $x_1 + 3x_2 \geq 6$
(5) $2x_1 + x_2 \geq 4$

Since the relationships of (4) and (5) are $\geq$, another non-slack variable must be introduced, x_6 and x_7, giving

(4) $-x_1 - 3x_2 - x_6 = -6$
(5) $-2x_1 - x_2 - x_7 = -4$

Now M = 5 and N = 4

Minimize

$-x_1 - 2x_2$

Subject to

(1) $y_3 - x_1 + 3x_2 = 10$
(2) $y_4 + x_1 + y_2 = 6$
(3) $y_5 + x_1 - x_2 = 2$
(4) $y_8 + x_1 + 3x_2 - x_6 = 6$
(5) $y_8 + 2x_1 + x_2 - x_7 = 4$

* Ref: GARVIN, W.W, Introduction to Linear Programming, pg 45-46.

Sample Problem II(continued)

where x_3, x_4, x_5, are natural slacks
x_8, x_9 are artificial slacks
$x_i \geq 0 \quad i = 1,2,\ldots,9$

The identifiers on output will be as follows:

Variable	Identifier
y_3	1
y_4	2
y_5	3
y_8	4
y_9	5
x_1	6
x_2	7
x_6	8
x_7	9

Sample Problem II (continued)

input		output
0504/	M,N	8
0.0e0/		.80000001e 01
0.0e0/		9
0.0e0/		.40000002e 01
0.0e0/		3
0.0e0/	Z	.40000000e 01
-1.0e0/		7
-2.0e0/		.40000000e 01
0.0e0/		6
0.0e0/		.20000001e 01
-1.0e0/		-.99999999e 01
3.0e0/		
0.0e0/	(1)	
0.0e0/		
10.0e0/		
1.0e0/		
1.0e0/	(2)	
0.0e0/		
0.0e0/		
6.0e0/		
1.0e0/		
-1.0e0/	(3)	
0.0e0/		
0.0e0/		
2.0e0/		
1.0e0/		
3.0e0/	(4)	
-1.0e0/		
0.0e0/		
6.0e0/		
2.0e0/		
1.0e0/	(5)	
0.0e0/		
-1.0e0/		
4.0e0/		

Execution Time -- 12 Seconds

7. There are no ERROR stops.

9. Input is from paper tape reader and output is from the paper tape punch.

10. Input and Output Formats

The first line of input contains the numbers defining M and N. This is followed by the cost coefficients for the functional, one element to a line. The coefficients for the slacks are entered first followed by the coefficients for the non-slack variables. The cost coefficients for the slacks are ordered according to the row position of the unit in the unit column vector associated with that slack. The cost coefficients are followed by the elements of the row vectors describing the coefficients of the non-slack variables. These are ordered by vector in the same order as the cost coefficients for these variables.

Within each row vector the order is by restraint equation, so that k within a vector is associated with restraint k. The row vectors associated with the non-slack variables are followed by the column vectors describing the right hand side. All elements defining the r.h.s. vector must be greater than or equal to zero and all row vectors are entered one element to a line.

Thus, for a program with M restraints and N non-slack variables, it is required that M, N, (M+N) cost coefficients, and M row vactors, each with M elements, be entered. Artificial slacks are handled by the pricing out technique, that is, by associating with them as cost coefficients a positive number which is large relative to the other cost coefficients so that the artificial slacks will be forced out of the basis.

The computation, always minimizes. Therefore, to maximize a functional, multiply the original functional by a minus one during set-up.

The output consists of an identified and value for each variable in the optimal basic solution and the optimal value of the functional. A slack is identified by an integer from 1 thru M where the integer indicated the restraint associated with that slack. A non-slack variable is identified by an integer from (M+1) thru (M+N) where the integer is associated with the ordering of the inputs pertaining to the variable. Thus the ordering of the cost coefficients for each variable, slack and non-slack, determines the identifier for that variable.

```
C     LINEAR PROGRAM USING SIMPLEX METHOD
 19   FORMAT(2I2)
 20   FORMAT(E14.8  )
      DIMENSION D(431)
 31   CONTINUE
      READ 19,M,N
      N1 = N + 1
      M1 = M + 1
      L  = M + N
      IA = N1 ' M1
 /    J  =  IA + 1
      KA =  IA + L
      READ 20, (D(I), I =J,KA)
      IB = M'N1
      READ 20, (D(I), I=1,IB)
      I = IB +1
      DO  29  J = I, IA
 29   D(J) =  0.0
      DO  2   I = 1,L
      IX = KA + I
 2    D(IX) = I
      DO  25  I  = 1,M
      IC =(I-1)'N1
      ID = IA + I
      DO 25 J = 1, N1
      IX = IC + J
      TEMP = D(IX) ' D(ID)
      IE = IB +J
 25   D(IE) = D(IE) + TEMP
      DO  28   J= 1,N
      K = J + M  + IA
      IE = IB + J
 28   D(IE) =D(IE) -D(K)
 18   TEMP= 0.0
      L = 0
      DO 3  J =1,N
      IE = IB +J
      IF (D(IE) - TEMP ) 3,3,4
 4    L = J
      TEMP = D(IE)
 3    CONTINUE
      IF (L) 10,10,5
 5    TEMP = .99999999E31
      K=0
      DO 6 I= 1,M
      IC =(I-1)' N1
      ID =  IC + L
      IF  ( D(ID) ) 6,6,8
 8    IX = IC + N1
      TEMP1 = D(IX) / D(ID)
      IF( TEMP1 - TEMP ) 9,6,6
 9    TEMP = TEMP1
      K= I
 6    CONTINUE
```

```
      IF (K) 10,10,11
10    DO 30 I= 1,M
      IX = KA + I
      J = D(IX)
      PUNCH 19, J
      KB=(I-1)'N1 +N1
      PUNCH 20,D(KB)
30    CONTINUE
      PUNCH 20, D(IA)
      PAUSE 1111
      GO TO 31
11    IX = K+KA
      I = D(IX)
      J = M+L
      IX = J+KA
      IY = K + KA
      D(IY) = D(IX)
      D(IX) = I
      DO 12  I=1,M1
      J =(I-1)'N1 + L
      IX = I + IA
      D(IX) = D(J)
12    D(J)= 0.0
      J = (K-1)'N1 + L
      D(J) = 1.0
      DO 13   J =1,N1
      JA=(K-1)'N1 + J
      IX = IA + K
13    D(JA) = D(JA) / D(IX)
      DO 14  I=1,M1
      IF(K - I ) 15,14,15
15    DO 14  J=1,N1
      JA=(I-1)' N1 + J
      JB=(K-1) 'N1 + J
      IX = IA + I
      D(JA) = D(JA) - ( D(JB) ' D(IX) )
14    CONTINUE
      GO TO 18
      END
      END
```

J1.
IDENTIFICATION

Title: Equation Solver - Ident. AJ 1.01
Programmer: (Lockwood, Kessler and Bartlett, Inc.) September 1962

PURPOSE

This program will solve 30 linear equations with 30 unknowns. A modified Gaussian Elimination Method is used. The program has the ability to output the derived matrix before executing the back solution to solve for the unknowns. This is accomplished by letting ike be 01. If the derived matrix is not desired, let ike be 02.

The program outputs the final matrix, the element location, and the value; i.e., $A_{2,4} = 6628.44$ would be output as 2 4 6628.44.

The value of the unknowns are elements $A_{i,n+1}$ where i - 1,2,--n.

LIMITATIONS

The elements on the diagonal must not be zero.

EQUIPMENT AFFECTED

CDC 160-A, Flexowriter.

SOURCE PROGRAM

(See following page)

INPUT FORMAT

```
CR
SSNCR
DDDDD.DDCR
DDDDD.DDCR
   .
   .
DDDDD.DD
```

Where SS is the value for ike either 01 or 02
NN is number of unknowns
DDDD.DD is the element. Elements are entered by rows.

4. Source Program

```
C          EQUATION SOLVER
100        format(2i2)
101        format(f8.2)
102        format(2i2,f8.2)
           dimension a(30,30)
25         read 100,iko,n
           if(n-30)26,26,27
26         j=1
           nn=n+1
           do 1 i=1,n
           do 1 k=1,nn
1          read 101,a(i,k)
9          i=nn
19         a(j,i)=a(j,i)/a(j,j)
           if(i-j)17,17,18
18         i=i-1
           go to 19
17         if(n-j)4,4,3
3          k=j+1
8          i=nn
5          a(k,i)=a(k,i)-a(j,i)*a(k,j)
           if(i-j)23,23,24
24         i=i-1
           go to 5
23         if(n-k)7,7,6
6          k=k+1
           go to 8
7          j=j+1
           go to 9
4          go to (21,20),iko
20         do 22 i=1,n
           do 22 j=1,nn
22         punch 102,i,j,a(i,j)
21         i=n-1
15         k=i+1
           x=0.
12         x=a(i,k)*a(k,nn)+x
           if(k-n)10,11,11
10         k=k+1
           go to 12
11         a(i,nn)=a(i,nn)-x
           if(i-1)13,13,14
14         i=i-1
           go to 15
13         do 16 i=1,n,1
           do 16 j=1,nn,1
16         punch 102,i,j,a(i,j)
           stop
27         pause 1111
           go to 25
           end
           end
```

OPERATING PROCEDURE

(1) Load 160 FORTRAN A Compiler III
(2) Mount object program
(3) Clear and run
(4) Mount data tape
(5) Turn on punch
(6) Cycle run switch

Note: If NN = 30 program will halt with 1111 displayed in a register.

Correct data tape and cycle run switch.

SAMPLE INPUT	SAMPLE OUTPUT	
1 4		
1.00	1 1	1.00
1.00	1 2	1.00
1.00	1 3	1.00
1.00	1 4	1.00
5.00	1 5	1.00
1.00	2 1	.00
-1.00	2 2	1.00
1.00	2 3	.00
1.00	2 4	.00
1.00	2 5	2.00
1.00	3 1	.00
2.00	3 2	.00
-1.00	3 3	1.00
1.00	3 4	.00
1.00	3 5	3.00
2.00	4 1	.00
1.00	4 2	.00
1.00	4 3	.00
1.00	4 4	1.00
6.00	4 5	-1.00

K1.

IDENTIFICATION
Title: Spiralled Way Alignment - Ident. AK1.01
Programmer: (Lockwood, Kessler & Bartlett, Inc.), September 1962

I. Description of Problem

This program computes the alignment and curve data for a horizontal alignment containing spirals and circular curves in any combination.

The alignment is defined by coordinates of the P. I.'s or by intersecting the P. I. from coordinates of two known points and the bearings of the P. I.

The curves are defined by the radius or degree of curve and the lengths of the leading spiral and the trailing spiral.

Chord or arc definition may be used for degree of curve controlled by the program (with use of a jump switch setting on the computer).

Values of Radius below 50 feet are assumed to be degree of curve and those values of 50 feet or above are used as radius of curve.

All descriptive headings are typed out to facilitate use of output data.

This program also computes the station and coordinates of all detail points on the alignment.

II. Limitations

1. The number of curves which can be computed is unlimited.

2. Output is provided via on-line typewriter.

3. Input is limited to tape produced on off-line flexowriter.

III. Accuracy

Input and output of distances and coordinates are limited to 10 significant digits with a maximum of 4 decimal places. Angles are rounded to the nearest 0.01 of a second.

IV. Equipment Affected

1. 160-A with on-line typewriter.

2. Off-line flexowriter required to produce input data tape.

V. Memory Allocation

Loc.	0000-0081	Sin-cos s/r
	0082-0157	Arctan s/r
	0158-0211	Square root s/r
	0212-0309	P I Intersection
	0310-0399	Bearing and angle typeout
	0400-0527	Spiral Routine
	0530-0551	Station typeout
	0552-0661	Alpha Storage
	0680-0799	Numeric & Temporary Storage
	0800-1603	Main Program
	1700-1713	Input Storage
	1800-1813	Input Storage
	1900-1913	Input Storage
	1920-1937	Input Storage

VI. Operating Procedure

1. Margin and tabs
   Margin at 20
   All tabs down
   Output paper at 12

2. Load Sicom

3. Place the program

4. Established typewriter input mode

5. Type 52 0000 /
   Reads the program and stops

6. Jump switch 1
   On for chord definition of degree of curve
   Off for arc definition of degree of curve

7. Place Flexowriter input tape

8. Type x 01 0800 /
   Reads one block of input tape and computes

9. Output and stops at 5

10. For next problem
    Cycle Run
    Repeat from step 6

11. Error indications
    (1) Program types "Spiral Overlap" and amount of overlap in degrees, and stops at 3 which indicates overlapping spirals on the curve.

    (2) Program types "Curve Overlap" and amount of overlap in feet, and stops at 4 which indicates overlapping curves between P I's.

12. After inspection of error typeout computations may be resumed by:
    Cycle Run

VII Description of Input

Job No. (up to 8 alpha characters)

Date (up to 16 alpha characters)

Station at beginning of increments (up to 10 digits)

Increments on curves (up to 10 digits)

Increments on tangents (up to 10 digits)
If incremented stations are not required enter very large numbers for both curves and tangents.

North coordinate of origin point (up to 10 digits)
East coordinate of origin point (up to 10 digits)
Station of origin point (up to 10 digits)

For Each P I.

P I number (up to 8 alpha characters)
Radius or degree of curve (Rc or Dc)
Values below 50 are considered to be degree of curve
** Degree of curve entered as: DD. MMSSXX
Length of leading spiral (enter 0. if not used)
Length of trailing spiral (enter 0. if not used)
North coordinate of P I (if known)*
East coordinate of P I (if known)*
Repeat for each P I

* If P I coordinates are not known, enter in these locations the coordinates of a point on tangent, the bearing to the P I, the coordinates of a point on the next tangent and the bearing to the P I.

The program willautomatically intersect the two tangents and store the P I coordinates in proper locations.

To terminate the problem enter only the P I number, . (this number will not be typed out) and the coordinates of the terminus point.

** Degree of curve may be either arc or chord definition. (See operating instructions).

VIII. Description of Output

This program will type the following information:

1. Title of Program
2. Job No.
3. Date
4. Station and coordinates of origin and the bearing to the first P. I.
5. Curve No. of the first curve and the bearing to the next P. I.
6. Data for the leading spiral (if any)
7. Data for the trailing spiral (if any)
8. Curve Data for the circular portion of the curve consisting of:

   Radius (R)
   Degree of Curve (Dc)
   Internal Angle (Ic)
   Length of Curve (Lc)
   Tangent of Curve (Tc)
   Chord Distance (Cc)
   External (Ec)
   Middle Ordinate (Mc)

ALIGNMENT

JOB NO. LKB-02

DATE SEPT 1 1962 (Month Day Year)

Sta. Beg. of Increments 765+00.000

Increments on Curves 200.000

Increments on Tangents 500.000

ORIGIN

NORTH 539634.956

EAST 1815553.618

STATION 761+98.960

PI No.	Rc or Dc	Ls	Ls'	NORTH	EAST	BEARING	NORTH	EAST	BEARING
1	2864.79	350.	350.	541220.761	1814903.376				
2	3819.72	0.	0.	544771.570	1816081.018				
3				547051.251	1815707.978				

ENGINEERING DIVISION

BY ..... DATE .....

CHKD. BY ..... DATE .....

REVISED .....

LOCKWOOD, KESSLER & BARTLETT, Inc. SYOSSET, N. Y.

PROJECT .....

SUBJECT .....

SHEET NO. ..... OF .....

JOB NO. .....

FLEXOWRITER TAPE PREPARATION

;LKB 02	JOB No.
;Sept. 1 1962	DATE
76500.	STA. BEGINNING OF INCREMENTS
200.	INCREMENT ON CURVES
500.	INCREMENT ON TANGENTS
530634.956	ORIGIN - Y COORD
1815553.618	" - X COORD
76198.96	" - STATION
/;1	P.I. #
2864.79	RADIUS OF CURVE
350.	LEADING SPIRAL LENGTH
350.	TRAILING " "
541220.761	Y COORD OF P.I.
1814903.376	X " " "
/;2	P.I. #
3819.72	RADIUS OF CURVE
0.	LEADING SPIRAL LENGTH
0.	TRAILING "
544771.57	Y COORD OF P.I.
1816081.018	X " " "
/;3	
547051.251	Y COORD OF TERMINAL POINT
1815707.978	X " " " "
/	

520000/x010800/

A L I G N M E N T

Job N° LKB 02

Date Sept. 1 1962

ORIGIN	STATION	NORTH	EAST	BEARING
	761+98.960	539634.9560	1815553.6180	N 22° 17' 43.96 W
CURVE N° 1				N 18° 20' 54.01 E

LEADING SPIRAL

Ls= 350.0000 p = 1.7814 k = 174.9782 x= 349.8694 y= 7.1248
LT= 233.3789 ST= 116.7081 LC= 349.9419 θ= 3° 30' .00 φ= 1° 9' 59.87

TRAILING SPIRAL

Ls= 350.0000 p = 1.7814 k = 174.9782 x= 349.8694 y= 7.1248
LT= 233.3789 ST= 116.7081 LC= 349.9419 θ= 3° 30' .00 φ= 1° 9' 59.87

CIRCULAR CURVE

R = 2864.7900 Dc = 2° 00' .00 Ic= 33° 38' 37.99 Lc= 1682.1948 Tc= 866.1283
Cc= 1658.1312 Ec = 128.0681 Mc= 122.5879
Ts= 1236.6033 Ts'= 1236.6033 I = 40° 38' 37.98

POINT	STATION	NORTH	EAST
P-I	779+12.901	541220.7610	1814903.3760
T-S	766+76.297	540076.6071	1815372.5240
S-C	770+26.297	540403.0231	1815246.3810
C-S	787+ 8.492	542060.1707	1815189.2770
S-T	790+58.492	542394.4957	1815292.6500
CENTER OF CURVE		541326.0358	1817958.4040
ON TANGENT	765+00.000	539913.4900	1815439.4080
ON SPIRAL	768+00.000	540191.1805	1815325.8850
	770+00.000	540378.1664	1815254.9650
ON CURVE	772+00.000	540569.0585	1815195.4330
	774+00.000	540763.6375	1815149.3600
	776+00.000	540960.9564	1815116.9720
	778+00.000	541160.0540	1815098.4270
	780+00.000	541359.9602	1815093.8150
	782+00.000	541559.7011	1815103.1590
	784+00.000	541758.3037	1815126.4140
	786+00.000	541954.8002	1815163.4660
ON SPIRAL	788+00.000	542148.2689	1815214.0120
	790+00.000	542338.9669	1815274.2690

CURVE N: 2 N 9° 17' 36.02 W

CIRCULAR CURVE

R = 3819.7200 Dc = 1° 30' .00 Ic= 27° 38' 30.04 Lc= 1842.7791 Tc= 939.6864
Cc= 1824.9602 Ec = 113.8879 Mc= 110.5905
Ts= 939.6864 Ts'= 939.6864 I = 27° 38' 30.04

POINT	STATION	NORTH	EAST
P-I	815+62.889	544771.5700	1816081.0180
P-C	806+23.203	543879.6570	1815785.2110
P-T	824+65.982	545698.9226	1815929.2690
CENTER OF CURVE		545082.0794	1812159.6850
ON TANGENT	795+00.000	544353.9769	1815942.5210
	800+00.000	544828.5570	1816099.9180
	805+00.000	545303.1372	1816257.3150
ON CURVE	810+00.000	544242.5637	1815886.0060
	812+00.000	544438.7348	1815924.8370
	814+00.000	544636.6692	1815953.3470
	816+00.000	544835.8245	1815971.4580
	818+00.000	545035.6547	1815979.1230
	820+00.000	545235.6122	1815976.3180
	822+00.000	545435.1489	1815963.0520
	824+00.000	545633.7178	1815939.3610
ON TANGENT	825+00.000	545732.4940	1815923.7750
	830+00.000	546225.9313	1815843.0310
	835+00.000	546719.3685	1815762.2860
TERMINAL	838+36.297	547051.2510	1815707.9780

K1.

IDENTIFICATION
Title: Traverse Adjustment - Ident. AK1.02
Programmer: (Lockwood, Kessler & Bartlett, Inc.), May 1962

1. Description of Problem

For this program, a traverse is defined as a series of measured distances and directions, starting from a point of known coordinates, and terminating at a point of known coordinates.

The program provides for the output of either the unadjusted or adjusted traverse (according to Compass Rule).

Input of directions can be in the form of either bearings or north azimuths, and output of directions, regardless of input, can be either bearings or north azimuths.

The computed precision, length of traverse, Y error, X error and resultant error are printed first, giving the user the option of examining the results and interrupting computations if necessary.

The program provides alphanumeric identification and column headings.

The program computes the area in square feet and acres of the unadjusted traverse. The area is printed at the end of the unadjusted output.

2. Accuracy

The program accepts distances and coordinates accurate to 0.001 units and direction angles accurate to 0.01 seconds.

The computed distances and coordinates are rounded to 0.001 $\pm$ units, and the angles are rounded to 0.01 $\pm$ seconds.

3. Limitations

The traverse may contain 200 maximum courses and may be in any one of the four quadrants.

All input is via Flexowriter tape.

1. Description of Problem

For this program, a traverse is defined as a series of measured distances and directions, starting from a point of known coordinates, and terminating at a point of known coordinates.

The program provides for the output of either the unadjusted or adjusted traverse (according to Compass Rule).

Input of directions can be in the form of either bearings or north azimuths, and output of directions, regardless of input, can be either bearings or north azimuths.

The computed precision, length of traverse, Y error, X error and resultant error are printed first, giving the user the option of examining the results and interrupting computations if necessary.

The program provides alphanumeric identification and column headings.

The program computes the area in square feet and acres of the unadjusted traverse. The area is printed at the end of the unadjusted output.

2. Accuracy

The program accepts distances and coordinates accurate to 0.001 units and direction angles accurate to 0.01 seconds.

The computed distances and coordinates are rounded to 0.001 ± units, and the angles are rounded to 0.01 ± seconds.

3. Limitations

The traverse may contain 200 maximum courses and may be in any one of the four quadrants.

All input is via Flexowriter tape.

4. Equipment Affected

C D C 160-A, 161
Flexowriter

5. Memory Allocation

Loc	0000 - 0081	Sin-cos s/r
	0082 - 0157	Arctan s/r
	0158 - 0211	Square root s/r
	0212 - 0229	Station Typeout routine
	0230 - 0266	Bearing to azimuth routine
	0267 - 0353	Bearing-angle output routine
	0354 - 0401	Temporary storage
	0402 - 0437	Alpha storage
	0438 - 0465	Input storage
	0466 - 0499	Temporary storage
	0500 - 0734	Main program
	0800 - 1998	Input storage

6. Operating Procedure

(1) Margin and tabs
Margin at 20
Every other tab down
Output paper at 10

(2) Load Sicom-A (May 1962)

(3) Place the program

(4) Establish typewriter input mode

(5) Type
52 0000 /
Reads the program and stops

(6) Jump switch 1
On for punch output (punch on)
Off for type output

(7) Jump switch 2
On for adjusted traverse
Off for unadjusted traverse

(8) Place Flexowriter input tape

(9) Type
x 01 0500 /

Reads the data tape and computes

(10) Output
Stops at 4 for unadjusted traverse
Stops at 3 for adjusted traverse

(11) For the next problem
Cycle Run (stops at 5)
Cycle Run again (halts)
Repeat from step (6)

7. Description of Input and Output

Input is prepared on off-line Flexowriter.

Job number, traverse number, Y code, X code, point No. of 1st point, and all other point numbers (as identification) are entered as one alpha word. Date is entered as two alpha words.

Computation time is approximately 2 1/2 seconds per course for punch output and 12 seconds per course for typewriter output.

Job Number	: 1234-5	CR	1st Point No.	: A-100	CR
Traverse No	: TRAV A	CR	Y 1st Point	1361201.345	CR
Date	: AUG. 13 1962	CR	X 1st Point	635425.584	CR
I/O Code	(11), 12, 21, 22	CR	Y Last Point	1359342.865	CR
Starting Sta.	34567.89	CR	X Last Point	635508.431	CR/
Y Code	: (North) South	CR			
X Code	: (East) West	CR/			

SHEET 1 OF 1
PAGE 6

PROGRAM 0108

TRAVERSE ADJUSTMENT

COMPASS RULE

Output? ☑ Type SJ ϕ1 Off ☐ Tape SJ ϕ1 On

Adjustment? ☑ No SJ ϕ2 Off ☑ Yes SJ ϕ2 On

I/O Code 1=Bearing 2=Azimuth

Quad Code 1=N/E;2=S/E;3=S/W;4=N/W

No	Point No.	Distance	Direction	No.	Point No.	Distance	Direction
2	A-101	966.648	255° 35' 15"	52			° ' "
3	A-102	503.215	324 42 36	53			
4	A-103	453.171	335 52 00	54			
5	A-104	543.031	326 04 10	55			
6	1			56			
7				57			
8				58			
9				59			
10				60			
11				61			
12				62			
13				63			
14				64			
15				65			
16				66			
17				67			
18				68			
19				69			
20				70			
21				71			
22				72			
23				73			
24				74			
25				75			
26				76			
27				77			
28				78			
29				79			
30				80			
31				81			
32				82			
33				83			
34				84			
35				85			
36				86			
37				87			
38				88			
39				89			
40				90			
41				91			
42				92			
43				93			
44				94			
45				95			
46				96			
47				97			
48				98			
49				99			
50				100			
51				101			

FLEXOWRITER TAPE PREPARATION

;1234-5	JOB NO
;TRAV A	IDENT NO
;AUG. 13 1962	DATE
11.	INPUT - OUTPUT CODE
34567.89	STARTING STATION
;NORTH	COORDINATE SYSTEM
;EAST	
/;A-100	1 ST POINT NO (STARTING POINT)
1361201.345	Y COORD 1 ST POINT NO
635425.584	X "
1359342.865	Y COORD TERMINAL POINT NO
635508.431	X "
/;A-101	2 ND POINT NO
966.648	DISTANCE TO 2 ND POINT
255.3515	BEARING
;A-102	3 RD POINT NO
503.215	DISTANCE TO 3 RD POINT
324.4236	BEARING
;A-103	
453.171	
335.52	
;A-104	TERMINAL POINT NO
543.031	DISTANCE TO TERMINAL POINT
326.041	BEARING
/	

SAMPLE OUTPUT

x010500/

Job Nº 1234-5 Date AUG. 13 1962

UNADJUSTED TRAVERSE Nº TRAV A

PRECISION	LENGTH	Y ERROR	X ERROR	DIST.ERROR
20753.834	2466.065	.017	.117	.118

Nº	STATION	DISTANCE	DIRECTION	NORTH	EAST	POINT
1	345 +67.890			1361201.345	635425.584	A-100
		966.648	S 55°35 '15.00 E			
2	355 +34.538			1360655.048	636223.059	A-101
		503.215	S 24°42 '36.00 W			
3	360 +37.753			1360197.910	636012.702	A-102
		453.171	S 35°52 ' .00 W			
4	364 +90.924			1359830.668	635747.189	A-103
		543.031	S 26° 4 '10.00 W			
5	370 +33.955			1359342.884	635508.549	A-104

area	
sq.ft.	4366.500
acres	.100

x010500/

Job Nº 1234-5 Date AUG. 13 1962

ADJUSTED TRAVERSE Nº TRAV A

PRECISION	LENGTH	Y ERROR	X ERROR	DIST.ERROR
20753.834	2466.065	.017	.117	.118

Nº	STATION	DISTANCE	DIRECTION	NORTH	EAST	POINT
1	345 +67.890			1361201.345	635425.584	A-100
		966.613	S 55°35 ' 8.23 E			
2	355 +34.504			1360655.041	636223.013	A-101
		503.228	S 24°42 '44.31 W			
3	360 +37.732			1360197.899	636012.632	A-102
		453.186	S 35°52 ' 7.11 W			
4	364 +90.918			1359830.654	635747.098	A-103
		543.045	S 26° 4 '18.18 W			
5	370 +33.964			1359342.866	635508.431	A-104

K1.

IDENTIFICATION
Title: Lines and Circle - Ident. AK1.03
Programmer: (Lockwood, Kessler & Bartlett, Inc.), August 1962

I. Statemtne of Problem

This program consists of the following five independent parts. The known data may be located in any of 4 quadrants.

Part I Intersections of Two Lines

The coordinates of two points are given to define the slope of each straight line. It is required to compute; (1) the coordinates of intersection of two lines, and (2) distances from each given point to the intersection.

Part II Intersection of Line and Circle

The coordinates of two points on a straight line, the coordinates of the center of a circle, and the radius of the circle are given. It is required to determine the coordinates of the intersections of the line and circle.

Part III Distance and Bearing

The coordinates of two points on a straight line are known. It is required to compute; (1) the distance between the two points, and (2) the bearing of the line.

Part IV Height of Triangle

The coordinates of three corners of a triangle are given. It is required to determine, (1) the height of triangle, and (2) the distance from first corner to the foot of perpendicular.

Part V Tangent, Arc, and Segment

The radius of a circle and the angle at the center of the circle are given. It is required to compute, (1) the tangent distance, (2) the arc length, and (3) the area of segment.

II. Limitations

1. The input data with proper mathematical signs, for all five parts, may be located in any one of four quadrants.

2. Part II: When there is no intersection of the given line and circle, the radius of the circle will be typed out as an error indication.

3. Part III: The interior angle I at the center of a circle must be less than 180°.

III. Equipment Affected

CDC 160-A
CDC 161
Flexowriter

IV. Memory Allocation

1000 - 1007	All Parts
1008 - 1069	Part I
1070 - 1099	Part II
1100 - 1117	Input all Parts
1118 - 1119	Constant
1120 - 1161	Part II
1162 - 1183	Part III
1184 - 1306	Part IV
1307 - 1335	Part V
1336 - 1349	Constants and intermediate storage
1350 - 1381	All Parts
1400 - 1482	Sin-cos s/r
1500 - 1559	Angle conversion s/r
1600 - 1645	Radius to Bearing s/r
1700 - 1776	Arctan s/r
1800 - 1854	Square root s/r
1000 - 1854	Program Tape

V. Description of Input Data

An angle as the input data is entered as degrees, minutes and seconds with degrees as integers, in the following form:
DDD. MMSSXX

Computation time will not exceed more than five seconds for any Part.

Part I: Intersection of two Lines

Loc.	1100	y_1	Coordinate on 1st line
	1102	x_1	"
	1104	y_2	"
	1106	x_2	"
	1108	y_3	Coordinate on 2nd line
	1110	x_3	"
	1112	y_4	"
	1114	x_4	"

Part II: Intersection of Line and Circle

Loc.	1100	y_1	Coordinate on a line
	1102	x_1	"
	1104	y_2	"
	1106	x_2	"
	1108	y_c	Coordinate of the center
	1110	x_c	"
	1112	R	Radius of the circle

Part III: Distance and Bearing

Loc.	1100	y_1	Coordinate of 1st point
	1102	x_1	"
	1104	y_2	Coordinate of 2nd point
	1106	x_2	"

Part IV: Height of Triangle

Loc.	1100	y_1	Coordinate of 1st corner
	1102	x_1	"
	1104	y_2	Coordinate of 2nd corner
	1106	x_2	"
	1108	y_3	Coordinate of 3rd corner
	1110	x_3	"

Part V: Tangent, Arc, and Segment
Loc. 1100 R Radius of circle in ft.
1102 I Interior angle

VI. Description of Output Data

Part I: Intersection of two lines

N E Coordinates of intersection

d_1 Distance from 1st point to intersection on first line
d_2 Distance from 2nd point to intersection on first line
$d_1 + d_2$ Distance from 1st to 2nd point on first line

d_3 Distance from 3rd point to intersection on second line
d_4 Distance from 4th point to intersection on second line
$d_3 + d_4$ Distance from 3rd to 4th point on first line

Part II: Intersection of Line and Circle

N' E' Coordinates of one intersection

N'' E'' Coordinates of other intersection

Part III: Distance and Bearing

d N DD MM SS E

Part IV: Height of Triangle

h l b - l

h = Perpendicular distance from 3rd corner to base
l = Distance from 1st corner to foot of perpendicular
b = Distance from 1st corner to second corner

Part V: Tangent, Arc and Segment

T L A

T = Tangent length
L = Arc length
A = Segment area in sq. ft.

VII. Operating Procedure

1. Margin and tabs
   (1) Margin at 20
   (2) All tabs down
   (3) Paper at 0

2. Load Sicom

3. Place the program tape

4. Input mode?
   (1) Flex input,
   Cycle Run to read the program (Stop 3)
   (2) Type input,
   11/ to read the program (Stop 3)

5. Jump switch 2
   (1) On for type output
   (2) Off for Flex output (punch on)

6. Jump switch 1
   (1) On for type input, go to step 7
   (2) Off for Flex input, go to step 10
   (Provide 5/ at the end of the tape)

7. Cycle Run (halts for type input)

8. Type input data (1100 - 1114)
   Type / after the last tab or CR for each solution.

9. Output of each solution
   Repeat from step 8

10. Cycle Run (for Flex input)

11. Reads the Flex input tape and output for each solution.

12. To alter the condition of input or output mode for the next solution, change positions of switch 1 or 2 as in steps 5 and 6 during computations.

VIII. Sample Input

Part I: Intersection of Two Lines

```
221053.462      189725.723
220355.616      190441.971
220004.625      189804.966
220983.902      190941.195
0.              /
```

Part II: Intersection of Line and Circle

```
206791.463      290535.797
204302.986      290775.556
205468.028      290520.962
  1165.920            0.     /
```

Part III: Distance and Bearing

```
395326.986      640001.947
396178.421      640646.194
0.              /
```

Part IV: Height of Triangle

```
919904.111      634350.135
920112.1        634206.425
920172.861      634467.323
0.              /
```

Part V: Tangent, Arc and Segment

```
475.            11.1137
  0.            /
```

IX. Sample Output

Part I: Intersection of Two Lines

220460.690	190334.126

849.431
150.569
1000.000

698.574
801.426
1500.000

Part II: Intersections of Line and Circle

204329.672	290772.985
206633.561	290551.011

Part III: Distance and Bearing

1067.706	N 37 06 48 E

Part IV: Height of Triangle

249.184	154.488	98.320

Part V: Tangent, Arc and Segment

46.547	92.799	139.933

K1.
IDENTIFICATION

Title: Transformation of Coordinates - Ident AK1.04
Programmer: (Lockwood, Kessler & Bartlett, Inc.), September 1962

I. Statement of Problem

This program will rotate and translate the coordinate of a series of points which define an enclosed area of traverse, compute the bearings and distances between these points, and find the area enclosed.

The input consists of the angle of rotation between the given system and the desired system, the x and y coordinates of the starting point in the desired system, and the x and y coordinates of each point in the given system.

II. Limitations

1. The number of points which can be computed is limited to 198.
2. Angle of rotation may be clockwise or counter clockwise, up to 360°.

III. Accuracy

Input and output of distances and coordinates are limited to 10 significant digits with a maximum of 4 decimal places. Angles are computed to 0.01 decimal places.

IV. Equipment Affected

1. 160-A with on line typewriter.
2. An off-line flexowriter may be used for input or output.

V. Memory Allocation

Loc	0000-0081	Sin-cos s/r
	0184-0237	Square root s/r
	0239-0314	Arctan s/r
	0318-0319	Alpha storage
	0320-0331	Input storage
	0332-0343	Temporary storage
	0344-0351	Input storage
	0352-0389	Alpha storage
	0394-0399	Numeric storage
	0400-0745	Main program
	0800-1999	Input storage

(3) Input data that is repeated for each solution

a. Parcel number (up to 8 alpha characters) CR /
b. Angle of rotation;
   if clockwise
   - if counter clockwise (up to 10 digits) CR
c. y coordinate of starting point in the desired system. (up to 10 digits) CR
d. x coordinate of starting point in the desired systems (up to 10 digits) CR
e. Identification number of point (up to 8 alpha characters, but not more than 6 spaces, if less fill field of 6 with space code) CR
f. y coordinate of first point to be transformed (up to 10 digits) CR
g. x coordinate of first point to be transformed (up to 10 digits) CR

(4) Repeat steps e., f., and g., for each succeeding point.

(5) After type-in of last x coordinate

1. type 0. CR
2. type 0. CR
   Program computes solution and returns to type-in state ready for entry of data at step (3) for next solution.

7. Program halts

(1) End of solutions (For flex tape input)
1. Selective Stop 2 on will cause program to halt after each solution
2. 5 / on tape after last input data will halt computations

(For type input)
1. Program halts at end of each solution (see step 6. (5)).

(2) Error alarm halt
1. Stop 4 is used to indicate number of points limit has been exceeded.

VI. Operating Procedure

1. Margin and tabs
   (1) Margin at 15
   (2) All tabs down
   (3) Paper at 0

2. Load Sicom and set type in mode.

3. Load program
   (1) Type 52 0000 / (reads program and stops)
   (2) Type x 010400 / (stops at 4)
   (3) Set Selective Jump 2 on

4. Set input mode

   (1) For flex tape input: Selective Jump 1 off and place input data tape on photo-reader.
   (2) For typewriter input: Selective Jump 1 on
   (3) For input verification: Selective Jump 4 on

5. Set output mode

   (1) For typewriter output: program is now set
   (2) For tape output:
      1. Master clear and cycle run
      2. Type 12/
      3. Type x 010413 / (stop 4)

6. Read input and compute

   (1) Cycle run
      A. For flex tape input; program reads tape for one solution and computes. Input form of tape is the same as that described below for type input.
      B. For typewriter input (with Selective Jump 4 off)

   (2) Input for first parcel only; program stores this data for use with succeeding solutions
      Type
         1. Previous Area (up to 10 digits) CR
         2. Job number and date (up to 24 alpha characters) CR /

SAMPLE INPUT

0.	STARTING AREA	
;9156-03	JOB NUMBER	
; Sept	DATE	
; 25 1962		
/;77	PARCEL NUMBER	
/-18.192	ANGLE OF ROTATION	
6912.679	Y COORD	OF STARTING POINT IN THE DESIRED SYSTEM
30845.884	X COORT	
;73.5	POINT NO.	OF FIRST POINT (STARTING POINT) TO BE TRANSFORMED
12345.678	Y COORD	
54321.987	X COORD	
;74.2	POINT NO.	OF SECOND POINT TO BE TRANSFORMED
12467.128	Y COORD	
54344.15	X COORD	
;77.1		
11768.755		
55042.53		
;77.2		
11268.834		
55033.656		
;76.2		
11442.482		
54048.048		
;76.2a	POINT NO.	OF LAST POINT TO BE TRANSFORMED
12345.678	Y COORD	
54321.987	X COORD	
0		END CODES
0.		
5/		

SAMPLE [illegible]

x010400/

PARCEL NO. 77 JOB NO. 9156-03 Sept 25 1962

POINT	BEARING	DISTANCE	Y COORDINATE	X COORDINATE
73.5			6912.6790	38843.8640
	N 7 58 49.10 W	123.4557		
74.2			7034.9391	38826.7243
	S 63 19 21.03 E	987.6535		
77.1			6591.5142	39709.2399
	S 17 18 19.01 E	499.9998		
77.2			6114.1478	39857.9713
	S 81 40 39.95 W	1000.0002		
76.2			5969.4076	38868.5014
	N 1 29 46.23 W	943.5931		
76.2a			6912.6790	38843.8640
	DUE EAST	.0000		
73.5			6912.6790	38843.8640

AREA	sq. ft.	acres
	701267.5800	16.0988
	701267.5800	16.0988

K2.

IDENTIFICATION

Title: Single Profile - Ident. AK2.01
Programmer: (Lockwood, Kessler & Bartlett Inc.), November 1962

I. Statement of Problem

Given basic data to define the vertical alignment of a highway the program will compute all the required data and elevations at even incremented stations.

The program is divided into two parts. The first part computes grades, PVI stations and elevations, vertical curve lengths and stopping sight distance for the selected curve.

The program will stop after computing part I to allow any change to be made before beginning part II.

Part II computes the elevation at incremented stations. The increment on vertical tangents may differ from the increment on vertical curves. In addition to evenly incremented stations the program will compute and type out the high points and low points on the curve, the elevation of odd stations not covered by the increments, the Middle Correction at each PVI and all other data required to fully define a highway profile.

The grades and elevations can either be rounded or unrounded as required. Appropriate errors alarms are built into the program with proper identification.

II. Limitations

1. The maximum number of PVI's is 33.

2. All elevations must be entered as being above zero.

3. The maximum number of "Odd Stations" is 49.

4. The starting point must be on tangent.

5. Except for Sight Dist. this program will work in Metric System.

III. Accuracy

1. Grades are rounded to the nearest 0.01' or when unrounded are carried as 10 significant digits.

2. Elevations are rounded to nearest 0.01'.

3. Vertical Curves are rounded upward to next 50' when total length is under 1000'. When length is over 1000', curves are rounded upward to next 100'.

IV. Equipment Affected

CDC 160-A
CDC 161 (Typewriter)
Flexowriter

V. Memory Allocation

0000 - 0002	Program
0005 - 0059	Sq. Rt. Subroutine
0060 - 0117	Program Subroutines
0125 - 0199	Main Program Part I
0200 - 0267	Alpha Storage
0300 - 0493	Main Program Part I
0496 - 0886	Main Program Part II
1370 - 1399	Temporary Storage
1400 - 1499	Input Storage (Odd Sta.)
1500 - 1773	Storage
1774 - 1799	Input Storage (Starting Block)
1800 - 1999	Input Storage (PVI Data)
0000 - 0886	Program Tape

VI. Description of Input Data

Initial Data:

1774	Job No:	Alpha-numeric identification of Job, up to 8 characters.
1776	Date:	Up to 16 alpha-numeric characters but more than 8.
1780	K Crest:	The lengths of curves are computed by L = K A, where A is the algebraic difference in grades. Enter the proper constant, based on design speed if the program is to compute length of curve. Enter any No. if the lengths are given.
1782	K Sag:	See description above.
1784	Max. Grade:	The program distinguishes between entered grades or elevations by comparing with this value. Any number greater than this value is assumed to be an elevation.
1786	Start Typeout:	The program will not output any incremented stations until this station is reached. Enter a rounded station.
1788	Inc. on Tang.:	Stations and elevations on vertical tangents will be output at this interval.
1790	Inc. on Curves:	Stations and elevations on vertical curves will be output at this interval.
1792	Stop Type Out:	The program will stop output when this station is reached.
1794	Starting Station:	Station at which profile begins.
1796	Starting Elevation:	Elevation of beginning of profile.

Note: The starting point must be on a vertical tangent.

1798	Add Constant to Elevation:	This value will be added to all elevations. The ability to add (or subtract) a constant allows the user to run a parallel profile by changing just one value.

PVI Data:

1800	Station of PVI:	Enter zero if the station is to be computed by intersecting two grades. (see sample problems)
1802	Elevation or Grade:	The program accepts either elevations in feet or grades in %. All values less than the value in loc 1784 are assumed to be grades. Enter a negative elevation when it is necessary to hold the elevation regardless of the rounding of the grades. (see sample)

Note: When working with elevations near zero, it is necessary to add a constant to all input elevations and then enter the negative of the constant in loc 1798. The final elevations output will be the correct values. This same procedure will work when working with negative elevations. (Do not use negative elevations as input except as noted above.)

1804	Vertical Curve:	Enter length of vertical curve if known. Enter -1. if the program is to compute the proper length of curve. (See loc 1780 and 1782)
		Enter up to a total of 33 PVI's in a similar manner.

Note: For last point, enter only the station and elevation then / (slash)

Odd Stations:

1400 - 1496 Enter up to 49 Odd Stations
If there are no odd stations enter O. CR/

VII. Description of Output Data

The output is in two parts. Part I types the curve data in the following form:

Tab No., Grade to PVI, Station, Elevation, Curve length and Sight distance.

Part II computes and outputs the incremented stations and the corresponding elevations. Odd stations, highpoints, lowpoints and all curve data required to define the vertical profile are also output.

VIII. Special Storage

A. Storage of values computed in Part I

No.	Station	Elevation	Grade	V. C.
0	1500	1502	1504	1506
1	1508	1510	1512	1514
2	1516	1518	1520	1522
.				
.				
.				
33	1766	1768	1770	1772

Location = No. x 8 + (ADDR of No. 0).

Example: The location of the elevation of the 10th PVI = 10 x 8 + 1502 = 1582

B. Special Locations

(1) Loc 0060 Station typeout format, 42 0602
(2) Loc 0064 Elev. and MC typeout format, 42 0602
(3) Loc 0068 Elev. typeout format, 42 0602
(4) Loc 0072 Curve and sight distance typeout format, 42 0600
(5) Loc 0076 Grade typeout format, 42 0604
(6) Loc 0228 Contains the limit by which the elevations may vary. See Operating Procedure, Steps 21 & 22 The constant 0.01 is stored in this location..

(7) Loc 0254 This location contains the length of curve above which the computed lengths are rounded upwards by 100' rather than 50'. The constant 1000 is stored in this location.

Example: If the program computes a length of curve of 907' it will round to 950'. If the computed curve is 1120' the program will round to 1200'.

(8) Loc 0256 Contains the station, elevation and grade round off. The constant 0.005 is stored.

IX. Operating Procedure

1. Margin and tabs
   (1) Margin at 15
   (2) All tabs set
   (3) Paper at 0

2. Load Sicom and establish the type input mode

3. Place the program tape

4. Read the program
   52 0000 / (halts)

5. Stop switch 1
   (1) On to stop for corrections after reading input tape
   (2) Off to continue computations

6. Stop switch 2
   (1) On to stop after computations but before typeout of Part I
   (2) Off to typeout of Part I

7. Jump switch 2
   (1) On for non-rounding of grades and elevations
   (2) Off for rounding to nearest 0.01

8. Place Flex input tape and type x 01 0000 / (Stop 3)

Part I

9. Cycle Run (to read the input tape and compute)

10. Stop 1 for correction?
    (1) No correction
        Cycle Run to continue
    (2) Correct the data
        (a) Clear and Run
        (b) 50 ADDR /
        (c) Type corrections
        (d) Resume computations x 01 0001 /

11. Stop 2 to eliminate Part I?
    (1) To type out Part I
        Cycle Run
    (2) To by-pass type out of Part I
        (a) Clear and Run
        (b) Go to step 13 and to step 14 (3)

12. Output of Part I (Stop 3)

Part II

13. Jump Switch 1
    (1) On for Flex output of Part II
        (turn on punch)
    (2) Off for type output of Part II

14. To change any computed values of Part I, if not go to step 15
    (1) Clear and Run
    (2) 50 ADDR /
        (See the list of Special Storage)
    (3) Compute Part II
        x 01 0002 /
    Go to Step 16

15. Cycle Run (to compute Part II)

16. Output of Part II (halts after final station and elevation)

17. For next solution, repeat from Step 5

18. "ERROR too many PVI's" (Stop 4)
    When this indication is typed out, revise input data and start again.
    (1) Clear and Run
    (2) Repeat from Step 5

19. "ERROR too many Odd Station's" (Stop 4)
    When this indication is typed out, remove excess odd stations and start again.
    (1) Clear and Run
    (2) Repeat from 5

20. "Curve Overlap" (Stop 4)
    Part I will type out the amount of overlap, when one curve overlaps the previous curve.
    (1) Repeat from step 14, or
    (2) Cycle Run to continue

21. "ERROR" in PVI Elevation (Stop 4)
    If the elevation computed in part II for the PVI does not agree within previously defined limits with the PVI elevation, the amount of discrepency is typed out labeled ERROR.
    Cycle Run to resume computations.

22. "ERROR" in PVT Elevation (Stop 4)
    The PVT elevation is computed two different ways. If they do not agree, the same error indication as above is typed out.
    Cycle Run to resume computations.

23. Stop 4
    The Stop 4 is provided for other errors without error typeout. Check input data carefully and try again.
    (1) Clear and Run
    (2) Repeat from Step 5

HWL PROFILE (0405) Checked by S.G.

LOC	DESCRIPTION	VALUE
1774	Job No.	LKE 05
1776	Date	Nov. 13, 1962
1780	K Crest	300.
1782	K S.L.	180.
1784	Max. Grade (%)	4.
1786	Start Typeout (station)	150000.
1788	Increment on Tangent	1000.
1790	Increment on Curve	500.
1792	Stop Typeout (station)	162250.
1794	Starting Station	143771.71
1796	Starting Elevation	204.85
1798	Add Constant to Elevation	0.

Are grades to be rounded? ☑ YES Jump switch 2 off
☐ NO Jump switch 2 on

P.V.I. DATA

No.	LOC	STATION / ELEV. or GRADE / *VERT. CURVE
1	1800	143741.71
	1802	203.35
	1804	300.
2	1806	153150.
	1808	226.
	1810	-1.
3	1812	155900.
	1814	-3.
	1816	500.
4	1818	0.
	1820	0.5
	1822	300.
7	1824	0.
	1826	-0.5
	1828	0.
6.5	1830	160100.
	1832	118.6
	1834	300.
16	1836	162250.
	1838	-119.25
	1840	
8	1842	
	1844	
	1846	
9	1848	
	1850	
	1852	
10	1854	
	1856	
	1858	
11	1860	
	1862	
	1864	
12	1866	
	1868	
	1870	
13	1872	
	1874	
	1876	
14	1878	
	1880	
	1882	
15	1884	
	1886	
	1888	
16	1890	
	1892	
	1894	
17	1896	
	1898	
	1900	
18	1902	
	1904	
	1906	
19	1908	
	1910	
	1912	
20	1914	
	1916	
	1918	
21	1920	
	1922	
	1924	
22	1926	
	1928	
	1930	
23	1932	
	1934	
	1936	
24	1938	
	1940	
	1942	
25	1944	
	1946	

Enter / after last point

LOC	ODD STATIONS
1400	150164.318
1402	155177.624
1404	157566.327
1406	160353.181
1408	
1410	
1412	
1414	
1416	
1418	
1420	
1422	
1424	
1426	
1428	
1430	
1432	
1434	
1436	
1438	
1440	
1442	
1444	
1446	
1448	
1450	
1452	
1454	
1456	
1458	
1460	
1462	
1464	
1466	
1468	
1470	
1472	
1474	
1476	
1478	
1480	
1482	
1484	
1486	
1488	
1490	
1492	
1494	
1496	
1498	

Enter / after last sta.

*Enter -1. if unknow

11-13-62

FLEXOWRITER INPUT	DESCRIPTION OF INPUT
;LK3 05	Job No.
;Nov. 13 1962	Date
300.	K Crest
130.	K Sag
4.	Max Grade
150000.	Start Typcout
1000.	Inc on Tang.
500.	Inc on Curve
162250.	Stop Typcout
149791.71	Start Station
204.85	Start Elev.
0.	Add Constant
149941.71	PVI Sta.
203.35	Elev.
300.	V C
152150.	Sta.
226.	EL
-1.	V C (Computer to Design Curve)
155900.	Sta.
-3.	Grade
500.	V C
0.	Sta. (Computer to intersect PVI Sta)
.5	Grade (to PVI)
300.	V C
0.	—
-.5	Grade (From PVI)
0.	—
162100.	Sta.
118.6	Elev. (Elev. Must be given for PVI after Intersection)
300.	V C
162250.	Sta.
-119.25	Elev. (The - Sign means this Elev. Must be held)
/150104.318	
155977.624	Odd Station
157500.327	
160353.181	

Part I

PROFILE

Job Nº LMB 05

Date Nov. 13 1962

PI	GRADE	STATION	ELEVATION	VERTICAL CURVE		
0		149791.71	204.85			
1	-1.0000	149941.71	203.35	300		
2	1.0300	152150.00	226.10	1300	S D	633
3	-3.0000	155900.00	113.60	500		
4	.5000	159500.00	131.60	300	S D	171
5	-.5000	162100.00	118.60	300		
6	.4333	162250.00	119.25			

CDD STATION

1	150164.32
2	155977.62
3	157564.33
4	160352.18

Part II

STATION	ELEVATION	CURVE DATA			
		GRADE	-1.0000		
150000.00	204.23				
150001.72	204.00	GRADE	1.0300		
150104.32	205.04				
151000.00	214.25				
151500.00	219.40	V C	1300	S D	633
151832.26	221.11	HIGH POINT			
152000.00	220.58				
152150.00	219.55	PVI EL	226.10	M C	-6.54
152500.00	214.20				
152800.00	206.60	GRADE	-3.0000		
153000.00	200.60				
154000.00	170.60				
155000.00	140.60				
155650.00	121.10	V C	500		
155900.00	115.49	PVI EL	113.60	M C	2.19
155977.62	115.03				
156000.00	114.89				
156078.57	114.67	LOW POINT			
156150.00	114.85	GRADE	.5000		
157000.00	119.10				
157566.33	121.93				
158000.00	124.10				
159000.00	129.10				
159350.00	130.85	V C	300	S D	771
159500.00	131.23	HIGH POINT			
159500.00	131.23	PVI EL	131.60	M C	-.37
159650.00	130.85	GRADE	-.5000		
160000.00	129.10				
160353.18	127.33				
161000.00	124.10				
161950.00	119.35	V C	300		
162000.00	119.14				
162100.00	118.95	PVI EL	118.60	M C	.35
162110.71	118.95	LOW POINT			
162250.00	119.25	GRADE	.4333		
162250.00	119.25	FINAL STATION			

K4.

IDENTIFICATION
Title: Composite Beams - Ident. AK4.01
Programmer: (Lockwood, Kessler & Bartlett, Inc.), September 1962

1. Description of Problem

This program has been developed to design interior or exterior steel beams for simple span highway bridges. Basically, the program is divided into two parts. The first part computes a solution for interior beams; the second part computes a solution for fascia or exterior beams. Each part may te run independently of the other. Provision can be made to use this program for selecting "built-up sections" as well as rolled shapes.

These programs store the properties of all wide flange shapes from a 30WF 108 to a 36WF 300 inclusive, together with various combinations of bottom cover plates. These stored properties are the cover plate width, cover plate thickness, beam depth, beam area and moment of inertia of beams. The 160-A Computer automatically tests each beam and chooses one or more that will satisfy the design criteria. Since several satisfactory solutions may be obtained for each problem, the engineer may base his final selection on the total weight of beam which is one of the output items.

2. Limitations

The span length must be greater than 40 feet.

This program will handle "simple" spans only.

This program does not have the provision to handle a top cover plate or concrete haunch.

3. Equipment Affected

CDC 160-A, 161
Flexowriter

4. Memory Allocation

Loc		
	0046-0099	Square Root
	0100-0611	Interior program
	0612-1199	Fascia program
	1200-1245	Beam table storage
	1246-1247	Alpha storage
	1248-1279	Constants
	1280-1291	Fascia program
	1292-1299	Fascia program
	1300-1345	Beam table storage
	1346-1347	Alpha storage
	1348-1399	Input storage
	1400-1445	Beam table storage
	1446-1447	Alpha storage
	1448-1499	Temporary storage
	1500-1545	Beam table storage
	1546-1547	Alpha storage
	1548-1599	Temporary storage
	1600-1645	Beam table storage
	1646-1647	Constant
	1648-1699	Temporary storage
	1700-1745	Beam table storage
	1746-1747	Constant
	1748-1799	Alpha storage
	1800-1845	Beam table storage
	1846-1847	Constant
	1848-1865	Alpha storage
	1866-1899	Fascia program
	1900-1945	Beam table storage
	1946-1947	Constant
	1948-1999	Fascia program

5. Operating Procedure

(1) Margin and Tabs
Margin at 15
Tabs at 26,38,41,42,43,51,64,77,90,105
(2) Load SICOM
(3) Establish Typewriter I/O Mode
(4) Mount Program Tape
(5) Type 520046/
Reads Program and Stops
(6) Type 501348/
Enter Input Data per Input Sheet
(7) When Last Input Item has been Intered
Type X010100/ for Interior
Calculations or X010612/ for Fascia
Calculation

Notes:

1. Program Stops

a) If all the stored beams are tried and none are satisfactory or if only two are satisfactory and five solutions have been asked for, the computer will print "Exceed Table".

b) If the ratio of the span in inches to the depth of the composite beam is greater than 25, the computer will print "Too Big F".

c) If the ratio of the span in inches to the depth of steel is greater than 30, the computer will print "Too Big F".

d) If the stringer spacing exceeds 14 feet, the computer will print "Too Big M11" and halt.

2. The number of solutions per problem (now set at 5) may be changed as follows:
a) Interior beams
Change location 0106 to 0 14 00 DR
b) Fascia beams
Change location 0618 to 0 14 00 DR where DR equals 1 less than the number of desired solutions.
ie. For 3 solutions DR = 02
For 1 solution DR = 00

3. To compute for a specific beam size (i.e. to keep fascia depths equal) proceed as follows:
   a) Interior beams
      Change location 0108 to 1 12 00 DR
      0111 to 2 02 00 DR
   b) Fascia beams
      Change location 0620 to 1 12 00 DR
      0623 to 2 02 00 DR where
      DR is the "Line" corresponding to the desired beam (see table).

TABLE

Line	Beam	Line	Beam	Line	Beam
00	30WF108	16	36WF160	32	33WF220
02	30WF116	18	36WF170	34	36WF230
04	30WF124	20	30WF172	36	33WF240
06	33WF130	22	36WF182	38	36WF245
08	33WF132	24	30WF190	40	36WF260
10	33WF141	26	36WF194	42	36WF280
12	36WF150	28	33WF200	44	36WF300
14	33WF152	30	30WF210		

4. Plate Girder Design Procedure

   a) Interior girders

      1) Change command in location 0106 to 0 04 0000 (gives single solution)
      2) Change command in location 0331 to 0 64 0333 (will not print DDWFDDD)
      3) Enter cover plate thickness in location 1646 in inches
      4) Enter cover plate width in location 1600 in inches
      5) Enter area of girder in location 1300 in square inches
      6) Enter depth of girder in location 1400 in inches
      7) Enter moment of inertia of girder in location 1500 in inches

b) Fascia girder

1) Change command in location 0618 to 0 14 0000 (gives single solution)
2) Change command in location 0924 to 0 64 0926 (will not print DDWFDDD)

Repeat steps (3), (4), (5), (6), and (7) if fascia girder is not the same as the interior girders.

c) For interior and fascia

1) Make allowable minimum stress low (location 1394 = 5.0) and allowable maximum stress high (location 1396 = 50.0) to insure typeout.
2) Enter cover plate thickness. Change location 1246 to; (BS) (LC) by XXXX where XXXX is cover plate thickness (i.e. XXXX = 1.25, max. 3 numbers and decimal point).

COMPOSITE BEAMS

Job No. A-4620-S Bridge Throg Neck

Date 9/20/62 By EMG

Symbol	Description	Loc.	Value	Units
Δ	Cover plate length for fascia (1)	1348	0.0	-
+	Thickness of wearing surface	1350	2.5	in.
S	Stringer spacing	1352	8.25	ft.
L	Span Length	1354	68.0	ft.
d	Slab thickness	1356	7.0	in.
X	Effective slab width (2)	1358	84.0	in.
h	Height of stud shear connectors	1360	4.0	in.
Z	Ratio of stress 16/18	1362	.88889	-
FS	Factor of safety for shear conn.	1364	2.0	-
f'c	Ultimate concrete stress	1366	3000.0	psi.
n	Ratio Es/Ec	1368	10.0	-
i	Diameter of shear connectors	1370	.75	in.
Q	Roadway width - curb to curb	1372	28.0	ft.
H	Sidewalk curb height -above slab	1374	.96	ft.
H	Mall curb height -above slab	1376	0.0	ft.
W	Sidewalk width	1378	5.0	ft.
W	Mall width	1380	0.0	ft.
D	Overhang beyond C/L of Fascia	1382	2.5	ft.
N	Number of beams	1384	5.0	-
N	Number of intermediate diaphragms (3)	1386	3.0	-
W	Weight of diaphragms	1388	0.0	k/ft.
W	Weight of railing	1390	.05	k/ft.
O	Type of shear connectors (4)	1392	1.0	-
R	Allowable min. stress in steel	1394	5.0	ksi.
R	Allowable max. stress in steel	1396	18.5	ksi.
R	Allowable concrete stress	1398	1.0	ksi.

Notes:

1. For full length plate enter 1. ; for computed length plate enter 0.
2. Smaller of following: 1/4 of span, center to center of beams or 12 times slab thickness.
3. Can be 0, 1, 2, or 3.
4. For spirals enter 1; for studs enter 2.

520046/501348/0.0
2.5
8.25
68.0
7.0
84.0
4.0
.88889
2.0
3000.0
10.0
.75
28.0
.96
0.0
5.0
0.0
2.5
5.0
3.0
0.0
.05
1.0
5.0
18.5
1.0
x010100/

INTERIOR	BEAM SIZE	PLATE SIZE	TOP STRESS	BOT STRESS	CON STRESS	PL LENGTH
	36WF170	14 BY1.	14.233	18.226	.701	49.556
TENTHS	.000	.100	.200	.300	.400	.500
POINTS	.000	6.800	13.600	20.400	27.200	34.000
SHEARS	76.564	66.183	55.803	45.423	35.043	24.662
SPACING	5.560	6.432	8.418	10.342	13.405	19.047

CAMBER	1.346
ALOW DEFL	1.020
ACT DEFL	.594
WEIGHT	14.084

FINIS

x010612/

FASCIA	BEAM SIZE	PLATE SIZE	TOP STRESS	BOT STRESS	CON STRESS	PL LENGTH
	36WF150	14 BY1.	14.316	18.103	.678	50.668
TENTHS	.000	.100	.200	.300	.400	.500
POINTS	.000	6.800	13.600	20.400	27.200	34.000
SHEARS	71.121	61.372	51.622	41.873	32.123	22.374
SPACING	5.810	6.733	8.891	10.961	14.288	20.514

CAMBER	1.429
ALOW DEFL	1.020
ACT DEFL	.539
WEIGHT	12.771

FINIS

8. <u>References</u>

a) American Association of State Highway Officials (AASHO)

b) Steel Construction Manual of the American Institute of Steel Construction (AISC)

K5.
IDENTIFICATION

Title: Horizontal Adjustment - Ident. AK5.01
Programmer: (Lockwood, Kessler & Bartlett, Inc.), November 1962

I. Statement of Problem

It is desired to obtain map or grid coordinates of photogrammetric points, namely the horizontal pass points, as the results of stereotriangulation.

The map coordinates of horizontal control points are given. The instrument coordinates of the control points and the pass points are observed for a triangulated strip of controlled photography.

Two basic transformation equations expressed in the form of a polynomial of a second degree are based on the *theory of errors by Army Map Service. The coefficients of the equations are determined by the method of least squares for the given map coordinates and the observed instrument coordinates of the control points.

The map coordinates of pass points on the strip are then computed by the transformation equations for the observed instrument coordinates.

II. Limitations

1. The number of horizontal control points (CP) on a strip shall not be less than 3 or more than 40.

2. An infinite number of horizontal pass points (PP) on the strip may be processed in groups of 200 or less at a time.

3. The X and Y map coordinates may range from 0 to 10,000,000.00 units.

4. The instrument x and y coordinates may range from 0.01 to 100,000.00 units.

5. The instrument x and y coordinates can not be zero and the y coordinates can not be negative values.

6. A maximum number of 50 control points in any order may be entered at a time to eliminate or to restore the previous data for the analysis of residuals.

*Photogrammetric Engineering, Sept. 1953, p. 627.

7. Identification numbers of pass points may range from 1 to 9999, starting from any number for first pass point and in numerical sequence for others.

III. Accuracy

The error of computed map coordinates will be less than ± 0.005 units.

IV. Equipment Affected

CDC 160-A
CDC 161
Flexowriter

V. Memory Allocation

0200 - 0649	Program
0700 - 0753	Alpha Storage
0754 - 0759	Punch s/r
0760 - 0761	Coeff.
0766 - 0841	Temporary Storage
1000 - 1099	Temporary Storage
1200 - 1999	Input Storage of PP Data
1600 - 1999	Input Storage of CP Data
0200 - 0761	Program Tape

VI. Description of Input Data

Control Points:

The information of control points consists of the identification, instrument coordinates and map coordinates for each point. A sequence of the control points can be in any order.

- I Alpha-numeric identification of a control point, up to 8 characters
- x Instrument x coordinate in the direction of triangulation
- y Instrument y coordinate
- Y Map coordinate in feet or meters; northing
- X Map coordinate in feet or meters; easting

The above input data can be prepared on a CDC Flexowriter or entered through the typewriter 161 according to the following format.

; I_1 tab x_1 tab y_1 tab Y_1 tab X_1 CR
; I_2 -------
.
.
.
.
; I_i -------
/

Pass Points:

The information of pass points consists of the instrument x and y coordinates for each point. A sequence of the pass points can be also in any order.

- x Instrument x coordinate of a pass point in the direction of triangulation
- y Instrument y coordinate

The input data of pass points will be prepared on the CDC Flexowriter.

x_1 tab y_1 CR
x_2 tab y_2 CR
.
.
.
.
x_j tab y_j CR
/

An identification number of first pass point will be entered through the typewriter 161 at the time when the information of the pass points is processed. The identification numbers of other pass points will be in numerical sequence. The numbers may range from 1 to 9999.

VII. Description of Output Data

The information of output data consists of three parts. Part one is the listing of control points. Part two is the computation of residuals. Part three is the computed coordinates of pass points.

The final output data of pass points may be listed through the typewriter 161 or they may be punched on the Flexowriter tape.

VIII. Operating Procedure

1. Margin and tabs
   (1) Margin at 20
   (2) Tabs at 27, 38, 50, 65, 80, 90 and 95
   (3) Paper at 0

2. Load Sicom

3. Place the program tape

4. Input mode?
   (1) Flex input (1 light), go to step 5
   (2) Type input (2 lights), 11/ to read the program and go to step 6

5. Cycle Run (to read the program) (Stop 3)

Control Points

6. Jump switch 1
   (1) On for type input of CP data, go to step 7
   (2) Off for Flex input of CP tape, go to step 8

7. Type in CP data (1600-1998)

   ; I tab x tab y tab Y tab X CR
   .
   .
   .
   .

   Type / after the last CR (Stop 4)
   Place new page and go to step 10

8. Place CP tape

9. Cycle Run (to read the CP tape)

10. Output of CP data (stop 3)
    Types: jp sw 2 on to pp

Residuals

11. Jump switch 2
    (1) On to compute PP data, go to step 15
    (2) Off to eliminate CP

12. Cycle Run
    Types: type cp nos.

13. Type CP Nos. in any order (CP = 1 to 40)
    - CP. tab (to eliminate)
    CP. tab (to restore)
    Type / after the last tab or CR

14. Output of residuals (stop 3)
    Types: jp sw 2 on to pp
    Repeat from step 11

Pass Points

15. Cycle Run
    Types: jp sw 2 on to tape output
    place pp tape type 1st pp no.

16. Place PP tape

17. Type in 1st PP No. (4 digits or less)
    PPPP. tab / (stop 3)

18. Jump switch 2
    (1) On for Flex output, punch on and go to step 19
    (2) Off for type output, go to step 21

19. Cycle Run (to read the PP tape)

20. Output of PP data (stop 4)
    Go to step 25

21. Stop switch 1 (for type output)
    (1) On to stop at 25th PP per page
    (2) Off to ignore 25th PP

22. Place tracing paper and double CR

Sample Input

Control Point Data

I	x	y	Y	X
;2 21	2334.28	270.28	935740.97	195348.41
;2 20 a	2230.20	359.98	936670.51	195028.90
;2 20 b	2271.63	396.72	935746.00	195414.17
;2 pp 5	2252.78	436.79	937063.07	195445.13
;2 pp 6	2226.51	269.14	936122.43	194678.63
;1 09	2093.61	421.20	937539.37	194406.11
;1 08	1988.71	408.92	937841.39	193714.84
;1 07	1855.23	411.30	938337.48	192899.16
;1 06	1710.50	388.03	938709.78	191917.34

/

Pass Point Data

x	y
2372.28	401.58
2372.16	417.11
2334.28	270.28
2226.51	269.14
2237.35	350.72
2230.20	359.98
2271.63	396.72
2230.77	403.89
2252.78	436.79
2243.59	518.85
2095.89	409.00
2093.61	421.20
2108.84	271.06
2109.53	517.83
1970.66	415.54
1988.71	408.92
1968.70	313.58
1963.99	526.53
1851.62	315.15
1838.37	405.71
1855.23	411.30
1859.87	533.63
1717.96	513.00
1721.47	407.00
1710.50	388.03

/

CP	x	y	Y	X	
1	2334.28	270.28	935740.97	195348.41	2 21
2	2230.20	359.90	936670.51	195028.90	2 20 a
3	2271.63	396.72	936746.00	195414.17	2 20 b
4	2252.78	436.79	937063.07	195445.13	2 pp 5
5	2226.51	269.14	936122.43	194678.63	2 pp 6
6	2093.61	421.20	937539.37	194406.11	1 09
7	1988.71	408.92	937841.39	193714.84	1 08
8	1855.23	411.30	938337.48	192899.16	1 07
9	1710.50	388.03	938709.78	191917.34	1 06

CP	Y	X	Residuals
1	.74	-1.52	2 21
2	-.69	-.02	2 20 a
3	1.02	2.65	2 20 b
4	-.95	-.10	2 pp 5
5	.38	.10	2 pp 6
6	.24	.29	1 09
7	-.11	-.65	1 08
8	-1.79	-1.69	1 07
9	1.20	1.00	1 06

jp sw 2 on to pp
type cp nos. -3. -8. /

1	.44	-.51	2 21
2	-.51	.39	2 20 a
3			
4	-.40	.35	2 pp 5
5	.15	.61	2 pp 6
6	.45	.03	1 09
7	-.20	-1.18	1 08
8			
9	.10	.33	1 06

jp sw 2 on to pp
jp sw 2 on to tape output
place pp tape type 1st pp no. 1. / 1

Sample Output

PP	x	y	z	r
1	2372.28	401.58	935024.21	196056.05
2	2372.16	417.11	936520.43	196111.36
3	2334.28	270.23	935741.41	195347.89
4	2226.51	269.14	936122.58	194679.24
5	2237.35	350.72	936587.07	195040.06
6	2230.20	359.98	936669.99	195029.29
7	2271.63	396.72	936747.40	195417.42
8	2230.77	403.89	936938.96	195191.02
9	2252.78	436.79	937062.66	195445.48
10	2243.59	518.85	937602.27	195684.44
11	2095.89	409.00	937456.26	194376.35
12	2093.61	421.20	937539.82	194406.14
13	2108.84	271.06	936557.71	193960.12
14	2109.53	517.83	938079.33	194852.15
15	1970.66	415.54	937946.95	193625.83
16	1988.71	408.92	937841.18	193713.65
17	1968.70	313.58	937323.80	193247.55
18	1963.99	526.53	938656.94	193983.15
19	1851.62	315.15	937753.45	192529.58
20	1838.37	405.71	938361.06	192772.34
21	1835.23	411.30	938335.16	192896.70
22	1859.87	533.63	939075.04	193364.11
23	1717.96	513.00	939456.61	192411.22
24	1721.47	407.00	938788.01	192053.49
25	1710.50	388.03	938709.88	191917.67

K6.

IDENTIFICATION

Title: Soil Consolidation - Ident AK6.01
Programmer: (Lockwood, Kessler & Bartlett, Inc.), October 1962

1. Description of Problem

This program computes the settlement of "n" layers of soil with a maximum of 4 layers and a surcharge load placed in "n" number of lifts which could be totally or partially removed.

The program will also compute the settlement of each layer at various intervals of time after removal of surcharge.

A maximum of 6 offset points per station can be entered as input and the settlement due to the consolidation of the soil at each point under the offsets can be computed.

The program provides alphanumeric identification and column headings.

2. Limitations

A maximum of 4 different layers of soil can be entered per each offset. A maximum of 6 offsets can be entered per station.

Six Pn and Cv values must be entered per each layer. If the laboratory data is applicable to more than one station, the test codes in the input may be set equal and the data should not be repeated.

All input is via flexowriter tape.

3. Equipment Affected

CDC 160-A, 161
Flexowriter

1. Description of Problem

This program computes the settlement of "n" layers of soil with a maximum of 4 layers and a surcharge load placed in "n" number of lifts which could be totally or partially removed.

The program will also compute the settlement of each layer at various intervals of time after removal of surcharge.

A maximum of 6 offset points per station can be entered as input and the settlement due to the consolidation of the soil at each point under the offsets can be computed.

The program provides alphanumeric identification and column headings.

2. Limitations

A maximum of 4 different layers of soil can be entered per each offset. A maximum of 6 offsets can be entered per station.

Six Pn and Cv values must be entered per each layer. If the laboratory data is applicable to more than one station, the test codes in the input may be set equal and the data should not be repeated.

All input is via flexowriter tape.

3. Equipment Affected

CDC 160-A, 161
Flexowriter

4. Memory Allocation

Loc	0000-0081	Sin-cos s/r
	0082-0171	Log 10 X s/r
	0172-0235	10^x s/r
	0236-0311	Arctan s/r
	0312-0333	Station type out routine
	0334-0355	Integration routine
	0356-0394	Series routine
	0395-0422	Interpolation routine
	0423-0451	Delta routine
	0452-0459	Weight routine
	0460-0499	Conversion routine
	0500-0549	Alpha storage
	0550-0599	Temporary storage
	0600-0974	Program
	1500-1599	Permanent storage
	1700-1955	Input storage
	1978-1999	Routine storage

5. Operating Procedure

(1) Margin and tabs
Margin at 20
Every other tab down
Output paper at 10

(2) Load Sicom

(3) Place the program

(4) Establish typewriter input mode

(5) Type
52 0000 /
Reads the program and stops

(6) Place Flexowriter input tape

(7) Type
x01 0600 /
Reads the data tape and computes

6. Description of Input and Output

Job number and test number are entered as one alpha word. Date is entered as two alpha words.

Computation time is approximately 3 minutes per offset with typewriter output.

SOIL CONSOLIDATION INPUT FORM

FIELD NO. 1

JOB NO. 91556-44

DATE: Oct 19 1962

FIELD NO. 2

STATION: 10050

TEST CODE: UL-77

FIELD NO. 3 (NUMERIC, MAX 10 DIGITS)

L1 =	-52.	m;	Base width of surcharge at left of centerline (neg)
L2 =	52.	m;	Base width of surcharge at right of centerline (pos)
SL =	.5	—	Side Slope (%)
Hf =	1.	m;	Height of lift
Pf =	128.	lbs/cf;	Unit weight of surcharge
MT =	6.	—	Total number of lifts
ML =	4.	—	Number of lifts left in place
Hx =	2.	—	Drainage condition code
t_c =	1.	month;	Time interval of surcharge placement
t_r =	12.	month;	Time interval from MT to ML

FIELD NO. 4 (NUMERIC, MAX 10 DIGITS) Negative to the left of centerline

Offset 1 =	-50.	m;	Offset 4 =	10.	m;
Offset 2 =	-30.	m;	Offset 5 =	30.	m;
Offset 3 =	-10.	m;	Offset 6 =	50.	m;

FIELD NO. 5 (NUMERIC, MAX 7 DIGITS) Negative below sea level

	Offset 1	Offset 2	Offset 3	Offset 4	Offset 5	Offset 6
Z1	6.	5.	3.	2.	1.	0.
Z2	4.	3.	1.	1.	0.	-1.
Z3	2.	2.	-1.	0.	-1.	-2.
Z4	-2.	1.	-3.	-1.	-2.	-3.
Z5	-2.	1.	-3.	-1.	-2.	-4.

FIELD NO. 6 (NUMERIC, MAX 10 DIGITS)

	First Layer	Second Layer	Third Layer	Fourth Layer	UNIT
q	2.	2.	2.	2.	tons/m²
eo	.67	.73	.88	.90	—
Cc	.19	.20	.39	.44	ft²/tons
Po	.256	.256	.27	.29	tons/sq.ft.
Pn	.512	.512	.512	.512	tons/sq.ft
Pn	.768	.768	.768	.768	tons/sq.ft.
Pn	1.024	1.024	1.024	1.024	tons/sq.ft.
Pn	2.048	2.048	2.048	2.048	tons/sq.ft.
Pn	4.096	4.096	4.096	4.096	tons/sq.ft.
Pn	8.192	8.192	8.192	8.192	tons/sq.ft.
Cv	99.5	99.5	95.3	106.1	(cm²/sec) 10^{-4}
Cv	48.2	48.2	105.36	122.3	(cm²/sec) 10^{-4}
Cv	32.19	32.19	127.10	149.7	(cm²/sec) 10^{-4}
Cv	156.78	156.78	165.24	156.8	(cm²/sec) 10^{-4}
Cv	149.68	149.68	159.56	161.3	(cm²/sec) 10^{-4}
Cv	140.52	140.52	113.88	168.7	(cm²/sec) 10^{-4}

FIELD NO. 7

Hd = . m; length of Drainage Path

FLEXOWRITER TAPE PREPARATION

FIELD No 1

;9156-44 JOB No

;Oct. 19 1962 DATE

FIELD No 2

/10050.00 STATION

;ul-77 TEST CODE

FIELD No 3

/-52. L_1

52. L_2

.5 S_L

1. H_F

128. P_F

6. M_T

4. M_L

2. H_X

1. t_c

12. t_p

FIELD No 4

/-50. OFF 1 -30. OFF 2 -10. OFF 3 10. OFF 4 30. OFF 5 50. OFF 6

FIELD No 5

/6. Z_1

4. Z_2

2. Z_3

-2. Z_4

-2. Z_5

Z_{1-5} FIRST OFFSET

5.

3.

2.

1.

1.

Z_{1-5} SECOND OFFSET

CONTINUE FIELD No 5

3.
1.
-1.
-3.
-3.

Z_{1-5} THIRD OFFSET

2.
1.
0.
-1.
-1.

Z_{1-5} FOURTH OFFSET

1.
0.
-1.
-2.
-2.

Z_{1-5} FIFTH OFFSET

0.
-1.
-2.
-3.
-4.

Z_{1-5} SIXTH OFFSET

FIELD No 6

2. q; 1ST LAYER	2. q; 2ND LAYER	2. q; 3RD LAYER	2. q; 4TH LAYER
.67 e_o "	.73 e_o "	.88 e_o "	.9 e_o "
.19 C_c "	.2 C_c "	.39 C_c "	.44 C_c "
.256 P_o "	.256 P_o "	.27 P_o "	.29 P_o "

CONTINUE FIELD N° 6

LAYER 1	LAYER 2	LAYER 3	LAYER 4
.512 P_1	.512	.512	.512
.768 P_2	.768	.768	.768
1.024 P_3	1.024	1.024	1.024
2.048 P_4	2.048	2.048	2.048
4.096 P_5	4.096	4.096	4.096
8.192 P_6	8.192	8.192	8.192
99.5 CV_1	99.5	95.3	106.1
48.2 CV_2	48.2	105.36	122.3
32.19 CV_3	32.19	127.10	149.7
156.78 CV_4	156.78	165.24	156.8
149.68 CV_5	149.68	159.56	161.3
140.52 CV_6	140.52	113.88	168.7

REPEAT AS FIELD NO 2 FOR END CODE

/ 000.00 STATION

/ END OF FIELD CODE

x010600/

JOB Nº 9456-44 DATE Oct. 19 1962

S O I L C O N S O L I D A T I O N

TEST Nº u1-77 STATION = 100 + 50.00 OFFSET, = 50.000 LEFT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .799

LAYER Nº 2 SETTLEMENT = .028

LAYER Nº 3 SETTLEMENT = .376

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .800
LAYER Nº 2 SETTLEMENT = .029
LAYER Nº 3 SETTLEMENT = .385

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .800
LAYER Nº 2 SETTLEMENT = .029
LAYER Nº 3 SETTLEMENT = .385

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .800
LAYER Nº 2 SETTLEMENT = .029
LAYER Nº 3 SETTLEMENT = .385

SETTLEMENT AT TIME = 96 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .800
LAYER Nº 2 SETTLEMENT = .029
LAYER Nº 3 SETTLEMENT = .385

SETTLEMENT AT TIME = 192 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .800
LAYER Nº 2 SETTLEMENT = .029
LAYER Nº 3 SETTLEMENT = .385

TEST N° ul-77 STATION = 100 + 50.00 OFFSET = 30.000 LEFT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800

LAYER N° 2 SETTLEMENT = .001

LAYER N° 3 SETTLEMENT = .001

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800
LAYER N° 2 SETTLEMENT = .001
LAYER N° 3 SETTLEMENT = .001

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800
LAYER N° 2 SETTLEMENT = .001
LAYER N° 3 SETTLEMENT = .001

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800
LAYER N° 2 SETTLEMENT = .001
LAYER N° 3 SETTLEMENT = .001

SETTLEMENT AT TIME = 96 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800
LAYER N° 2 SETTLEMENT = .001
LAYER N° 3 SETTLEMENT = .001

SETTLEMENT AT TIME = 192 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .800
LAYER N° 2 SETTLEMENT = .001
LAYER N° 3 SETTLEMENT = .001

TEST Nº u1-77 STATION = 100 + 50.00 OFFSET = 10.000 LEFT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082

LAYER Nº 2 SETTLEMENT = .115

LAYER Nº 3 SETTLEMENT = .004

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082
LAYER Nº 2 SETTLEMENT = .115
LAYER Nº 3 SETTLEMENT = .004

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082
LAYER Nº 2 SETTLEMENT = .115
LAYER Nº 3 SETTLEMENT = .004

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082
LAYER Nº 2 SETTLEMENT = .115
LAYER Nº 3 SETTLEMENT = .004

SETTLEMENT AT TIME = 96 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082
LAYER Nº 2 SETTLEMENT = .115
LAYER Nº 3 SETTLEMENT = .004

SETTLEMENT AT TIME = 192 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .082
LAYER Nº 2 SETTLEMENT = .115
LAYER Nº 3 SETTLEMENT = .004

TEST Nº u1-77 STATION = 100 + 50.00 OFFSET = 10.000 RIGHT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000

LAYER Nº 2 SETTLEMENT = .000

LAYER Nº 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000

TEST N° ul-77 STATION = 100 + 50.00 OFFSET = 30.000 RIGHT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000

LAYER N° 2 SETTLEMENT = .000

LAYER N° 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000
LAYER N° 2 SETTLEMENT = .000
LAYER N° 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000
LAYER N° 2 SETTLEMENT = .000
LAYER N° 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000
LAYER N° 2 SETTLEMENT = .000
LAYER N° 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 96 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000
LAYER N° 2 SETTLEMENT = .000
LAYER N° 3 SETTLEMENT = .000

SETTLEMENT AT TIME = 192 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER N° 1 SETTLEMENT = .000
LAYER N° 2 SETTLEMENT = .000
LAYER N° 3 SETTLEMENT = .000

TEST Nº ul-77 STATION = 100 + 50.00 OFFSET = 50.000 RIGHT

SETTLEMENT AT TIME OF REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000

LAYER Nº 2 SETTLEMENT = .000

LAYER Nº 3 SETTLEMENT = .000

LAYER Nº 4 SETTLEMENT = .022

SETTLEMENT AT TIME = 12 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000
LAYER Nº 4 SETTLEMENT = .022

SETTLEMENT AT TIME = 24 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000
LAYER Nº 4 SETTLEMENT = .022

SETTLEMENT AT TIME = 48 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000
LAYER Nº 4 SETTLEMENT = .022

SETTLEMENT AT TIME = 96 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000
LAYER Nº 4 SETTLEMENT = .022

SETTLEMENT AT TIME = 192 MONTHS AFTER REMOVAL OF SURCHARGE

LAYER Nº 1 SETTLEMENT = .000
LAYER Nº 2 SETTLEMENT = .000
LAYER Nº 3 SETTLEMENT = .000
LAYER Nº 4 SETTLEMENT = .022

L1.

IDENTIFICATION

Title: Statistical Program I (Means, Standard Deviations, Correlation for Two Variables) - Ident. L1.01
Contributor: Dr. C. A. Caceres, August 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the 2 x $_n$ array would allow running on the 160). The routine produces the means and standard deviations for two sets of variables and the correlation between corresponding entries in the two lists.

USAGE

1. Operational Procedure:

   Using techniques for FORTRAN the integral value of n and the n-floating point x-values and the n-floating point y-values are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find his results well-labelled.

   To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

   The maximum array size for which the routine is written is 2 x 400_{10}.

10. Input and Output Formats:

    Two format statements are used to control the input of data into the routine:

```
 4 FORMAT (i4)
10 FORMAT (f12.6)
```

    Three read commands are given with pauses between them. The first read command accepts an integral quantity n (xxx/), and the next two read commands read in successively n x-values and n y-values.

    Note: It has been made the practive to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160 or 160-A Computer, and Flexowriter.

L1.

IDENTIFICATION

Title: Statistical Program II (linear Regression) - Ident. L1.02
Contributor: Dr. C. A. Caceres, August 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the 2 x n array would allow running on the 160). The routine produces the following quantities for the two sets of variables:

Regression coefficients, alpha and beta
The sum of squares attributable to regression
The sum of squares of deviations from regression
The mean square deviation from regression
The sample standard error of the dependent variable estimated from the independent variable
The sample standard error of beta and,
The 't-test' value

Reference: SNEDECOR, Statistical Methods, Iowa State College Press

USAGE

1. Operational Procedure:

Using the techniques for FORTRAN the integral value of n and the n-floating point x-values and the n-floating point y-values are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find the results well-labelled.

To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

The maximum array size for which the routine is written is 2 x 400_{10}.

10. Input and Output Formats:

Two format statements are used to control the input of data into the routine:

```
100 FORMAT (i4)
101 FORMAT (f12.6)
```

Three read commands are given with pauses between them. The first read command accepts an integral quantity n (xxx/) and the next two read

commands read in successively in x-values and n y-values.

Note: It has been made the practive to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160 or 160-A Computer, and Flexowriter.

L1.

IDENTIFICATION

Title: Statistical Program III (Test of Nomality in a Frequency Distribution of Large Size) - Ident L1.03
Programmer: Dr. C. A. Caceres, August, 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the line array would allow running on the 160). The routine produces the following quantities:

s_1, s_2, s_3, s_4, S_2, S_3, S_4, k_1, k_2, k_3, k_4, g_1, g_2, $s_{g_1}^2$, $s_{g_2}^2$, s_{g_1}, s_{g_2}, t for g_1 and t for g_2.

Reference: SNEDECOR, Statistical Methods, Iowa State College Press

USAGE

1. Operational Procedures:

   Using techniques for FORTRAN the integral value of n and the n-floating point x-values are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find his results well-labelled.

   To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

   The maximum array size for which the routine is written is $1 \times 800_{10}$.

10. Input and Output Formats:

    Two format statements are used to control the input of data into the routine:

    ```
    100 FORMAT (i4)
    101 FORMAT (f12.6)
    ```

    Two read commands are given with a pause between them. The first read command accepts an integral quantity n (xxx/) and the second reads in n values of the variable from paper tape.

    Note: It has been the practice to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160-A or 160 Computer, and Flexowriter.

L1.

IDENTIFICATION

Title: Statistical Program IV (Mean and Standard Deviation for One Variable) - Ident. L1.04
Contributor: Dr. C. A. Caceres, August 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimention size of the line array would allow running on the 160). The routine produces the mean and standard deviation for a single variable.

USAGE

1. Operational Procedure:

Using the techniques for FORTRAN the integral value of n and the n-floating point x-values are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find his results well-labelled.

To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

The maximum array size for which the routine is written is $1 \times 1000_{10}$.

10. Input and Output Formats:

Two format statements are used to control the input of data into the routine:

```
4 FORMAT (i4)
10 FORMAT (f12.6)
```

Two read commands are given with a pause between them. The first read accepts an integral quantity n (xxx/) and the second reads in n values of the variable from paper tape.

Note: It has been made the practive to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160 or 160-A Computer, and Flexowriter.

L1.

IDENTIFICATION

Title: Statistical Program V (Sums, Sums of Squares, and Sums of Cross Products for Three Variables) - Ident. L1.05
Contributor: Dr. C. A. Caceres, August 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the array would allow running on the 160). The routine produces the basic statistical building blocks for a three-dimensional sample-space, i.e., sums, sums of squares and sums of cross products. This routine would reduce the dog-work for a skilled statistician and enable him to rapidly decide how best to handle his data.

USAGE

1. Operational Procedure:

   Using Techniques for FORTRAN the integral value of n and the n-floating point values of each of the three variables are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find the results well-labelled.

   To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

   The maximum array size for which the routine is written is $3 \times 300_{10}$.

10. Input and Output Formats:

    Two format statements are used to control the input of data into the routine:

```
112 FORMAT (i4)
113 FORMAT (f12.6)
```

    Four read commands are given with three pauses between them. The first read command accepts an integral quantity n (xxx/) and the other three read in n values of each variable from paper tape.

    Note: It has been made the practive to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum 160 or 160-A Computer, and Flexowriter.

L1.

IDENTIFICATION

Title: Statistical Program VI (Percentiles) - Ident. L1.06
Contributor: Dr. C. A. Caceres, September 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the four internal storage arrays would allow running on the 160). The routine produces the percentiles at 5 percent intervals after sorting the original data internally.

USAGE

1. Operational Procedure:

   Using techniques for FORTRAN the integral value of n and the n-floating point x-values are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find his results well labelled.

   To continue theprogram with new data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

   The maximum data array size for which the routine is written is $1 \times 300_{10}$.

10. Input and Output Formats

    Two format statements are used to control the flow of data into the routine:

    ```
    103 FORMAT (i4)
    104 FORMAT (f12.6)
    ```

    Two read commands are given with a pause between them. The first read command accepts an integral quantity n (xxx/), and the next read command reads in the n x-values.

    Note: It has been the practice to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160 or 160-A Computer, and Flexowriter.

METHOD

The necessary sorting of the data is carried out by means of the "string" sorting technique in which strings of ascending values are identified and the strings lengthened by merging in successive passes until one sorted string results. Besides the two equal-sized areas used for handling the actual floating point data, two integral areas are used to store the starting and ending addresses of the lth string during each merge. As soon as a pass is terminated in which i has not exceeded unity, the sort is known to be completed and the relatively simple process of computing the percentiles can be undertaken.

L1.

IDENTIFICATION

Title: Statistical Program VII (Two 't'-tests for 2 Equal or Unequal-sized Line Arrays) - Ident. L1.07
Contributor: Dr. C. A. Caceres, August 1962

PURPOSE

This program is written in FORTRAN for the Control Data 160-A (reducing the dimension size of the line arrays would allow running on the 160). The routine produces and outputs either the t-test value for equal-sized sets or the t-test value for unequal-sized sets, depending on whether the sets are equal or unequal in length. (Also output are intermediate statistical quantities of interest produced during computation.)

Reference: Snedecor, Statistical Methods, Iowa State College Press

USAGE

1. Operational Procedure:

Using techniques for FORTRAN the integral value of n_1, the n_1-floating point x-values, the integral value of n_2 and the n_2-floating point y-values, are read in from paper tape. The resulting tape from the punch is for listing on the Flexowriter and the user will find his results well-labelled.

To continue the program with fresh data it is not necessary to reload the program; simply place the new data tape(s) in the reader and run.

3. Space Requirements:

The maximum array size for which the routine is written is $2 \times 400_{10}$.

10. Input and Output Formats:

Two format statements are used to control the input of data into the routine:

```
109 FORMAT (i4)
110 FORMAT (f12.6)
```

Two read commands are given followed by a pause and two more read commands. The first two read commands accept an integral quantity n_1 (xxx/) and the n_1 x-values; the second two read commands bring in n_2 and the n_2 y-values.

Note: It has been the practive to oversize the format statement and thus render it mandatory to terminate each piece of data with a slash (and, if desired, carriage return).

14. Equipment Configuration: Minimum SWAP 160-A or 160 computer, and Flexowriter.

Z1.

IDENTIFICATION

Title: Single Precision Fractional Square Root - Ident. Z1.00
Programmer: Sanford Elkin, February 1961

PURPOSE

This is a demonstration routine. It will find the square root of a proper fraction with a maximum error of 2^{-11}.

USAGE

1. Operational Procedure:

   Load the bi-octal tape starting at cell 0, and clear. Place the number N in the A register, and start. $X = \sqrt{N}$ will appear in the A register. Place a new N in A and repeat.

3. Space Used: Octal cells 100-206, plus locations 0, 1, 7, and 55-67.

10. Timing: Each interation about 2.25 milliseconds, maxi ım of 25 milliseconds.

11. Accuracy: Maximum error is 2^{-11}.

MATHEMATICAL METHOD

Newton-Raphson interation with $X_{i+x} = 1/2\ (X_i + N/X_i)$ and $X_0 = 1$, stopping when $\Delta X \leq 2^{-11}$. The Single precision Divide Subroutine is used.

Z1.

IDENTIFICATION

Title: Random Number Generator (12 bit) - Ident. Z1.02
Programmer: J. A. Pederson, July 1961

PURPOSE

Produce 12-bit "random" numbers

USAGE

1. Operational Procedure:

   The routine floats and may be loaded anywhere. Set P = any arbitrary value, set Load switch and Run.

   There are two entrances into the routine. A reset entrance at L and a random number entrance at L + 10_8 (where L = the initial address of the routine).

   The initial entry is made at L with an exit with the random number displayed in A. Entry again at L restarts the sequence.

3. Space Required: $46_8 = 38_{10}$ locations

4. Temporary Storage: None

10. Timing: 307 microseconds per number generated.

12. Cautions to User: The routine generates a cycle of 1679 X 1678 2,560,000 12 bit random numbers without repeating the sequence. Beyond this the cycle repeats itself.

METHOD

The routine generates two number series:

$$a_i + 2 = (a_{i+1} + a_i) \bmod 2^{12} - 1$$

$$b_i + 2 = (b_{i+1} + b_i) \bmod 2^{12} - 1$$

where

$$a_1 = 1, \; a_2 = 2$$

$$b_1 = 1, \; b_2 = 100_8$$

Each series has a cycle of 1679 values before repeating. The "a" series is shortened to a 1678 cycle by a program stop. The random number is obtained by an exclusive "or" of a_i and b_i.

Z3.
IDENTIFICATION

Title: ALNUP - Ident. Z3.00
Programmer: C. M. Atchison, April 1961

PURPOSE

This routine will punch character messages in paper tape that are legible to an unskilled observer. These characters are formed in a 5 by 7 matrix on an output tape. The standard 160 Flexowriter coded paper tape is used as input.

USAGE

1. Operational Procedure:

   a. Prepare Flexowriter (Standrad CDC 160 code) input tape.
   b. Load BI-OCTAL program tape at location zero.
   c. Turn punch on.
   d. Place Flex input tape in reader.
   e. Run from zero.
   f. Normal stop: P = 140, Z = HLT 77.

3. Space Required
   a. 574_8 locations.
   b. 380_{10} locations.

4. Temporary Storage Requirements
   a. Variable according to length of input file starting at location 600_8.

5. Cautions to User
   a. The program does not punch leader or trailer on the output paper tape.

13. Equipment Configuration
   a. Standard 160 coded Flexowriter for preparation of input tape.
   b. 160 computer with paper tape input and output.

Z3.

IDENTIFICATION

Title: Trajectory Calculation - Ident. AZ3.02
Programmer: Irv. Hecker, August 1961

PURPOSE

Given the target range, the velocity of the projectile, and its angle of inclination, the program will plot the path of the projectile on the typewriter. Appropriate comments are typed to indicate whether the shot hit the target, was too short, or too long. A shot is scored as a hit if it falls within 10.0 feet of the range.

The plot is scaled so that each unit on the X-axis is 1/80 of the distance to the target. Each unit on the y-axis is 1/80 of the maximum height allowed. The maximum allowable height is 1/4 of the target range.

USAGE

1. Operational Procedure

   a. Place the program tape in the paper tape reader

   b. Turn PTR on.

   c. Master Clear

   d. Set the relative bank control to the number of the bank into which the program is to be loaded.

   e. Place "Load-Clear" switch on "Load".

   f. Place "Run-Step" switch on "Run". Tape will be read into the computer memory. Check sum is 0001.

   g. Master Clear

   h. Set all bank controls (buffer not necessary) to the number of the bank in which the program is.

   i. Place "Run-Step" switch on "Run".

9. Input and Output Formats

   a. Entering Data:

The program will request a value for "RANGE(FT)", "VELOCITY (FT/SEC.)" and "ANGLE (DEG.)".

Each value is entered as a fixed point number with up to nine significant digits. A decimal point must be entered in the proper position within the number. Each value must be followed by either a tab or a carriage return. Zero must be entered as "0.0".

b. Output Format:

The routine sets up the output graph so that the scale in the direction of the target is one space equals 1/80 of the range, and one space vertically equals 1/320 of the range. The maximum height attainable by trajectory which will stay on the page is 1/4 of the range. The numerical quantities printed have only two digits to the right of the decimal point.

13. Equipment Configuration

Minimum 160-A and 161 Typewriter

METHOD

The coordinates of the projectile are computed by the equation:

$$Y_1 = X_1 \tan 0 - g X_1^2/2 Vo^2 \cos^2 0$$

where 0 = initial angle of inclination

VO = initial velocity

g = acceleration due to gravity

i = 1, 2, 3, ... , 16

and since X_i is computed every five units, $X_i = \frac{5r}{80} . i$

where r = target range.

Z3.
IDENTIFICATION

Title: MORTGAGE AMORTIZATION - Ident. AZ3.03.
Programmer: (Control Data Corporation), July 1962

PURPOSE

This program has two options:

1. Computes the monthly payment necessary to amortize a given principal amount at a given interest rate for a given duration.

2. Computes the duration necessary to amortize a given principal amount at a given interest rate with a given monthly payment.

In option (2), after computing the necessary duration of the loan, which will generally be a non-integral number of years, a desired duration is requested (some practical value for the number of years) and new monthly payment, close to the desired amount, is calculated.

At the completion of either option, a complete amortization schedule is typed out, showing the payment number, the portion of the payment which goes for interest, the protion which is deducted from the principal and the new balance to be paid.

USAGE

1. Loading instructions

   a. Place the program tape in the paper tape reader.
   b. Turn PRT on.
   c. Master clear.
   d. Set the relative bank control to the number of the bank into which the program is to be loaded.
   e. Place "Load-Clear: switch on "Load."
   f. Place "Run-Step: switch on "Run". Tape will be read into the computer memory. Check sum is 0002.
   g. Master clear.
   h. Set all bank controls (buffer not necessary) to the number of the bank in which the program is contained.
   i. Place "Run-Step" switch on "Run".

2. Operating Instructions:

   The program will request the principal amount of the loan and the rate of interest. Each number must contain a decimal point and following digit. When the values for these two quantities have been supplied,

the program will type "Desired Monthly Payment = ". If a certain monthly payment is desired, the amount is specified at this point, and the program will compute the number of years necessary to amortize the loan. If no specific monthly payment is desired, type "0.0" for the desired monthly payment.

The program will request "Desired Duration of Loan = ". At this point the length of the loan in years must be supplied. The program will compute the monthly payment necessary to amortize the loan. Following this, a payment schedule is typed showing payment number, interest due, amount paid on principal, and balance on loan.

7. Error Stop

If a value is input for desired monthly payment which is too small, the computer will type "Sorry, not possible" and halt. To restart, place run switch in neutral and back to run.

14. Equipment Configuration - Minimum SWAP 160-A, 161 typewriter.

METHOD

The equation used to calculate the monthly payment is:

$$M = \frac{P.\ R.\ (1 + R)^N}{(1 + R)^N - 1}$$

where P = principal
R = rate of interest per month (1/12 annual rate)
N = number of months

The equation used to calculate the duration of the loan in years is:

$$D = -\frac{\log\left(1 - \frac{PR}{M}\right)}{\log(1 + R)}$$

where M = monthly payment and R and R are defined as above.

B1.

IDENTIFICATION

Title: Multiple Precision Package - Ident. B1.02
Programmer: D. C. Nelson, May 1961

PURPOSE

Perform arithmetic with operands expressed to 6, 12, 18, 24, ...6N binary digit precision.

USAGE

Normally the degree of precision will be set for a particular tape, but this can be changed for other precision without altering any of the subroutines. By specifying the beginning and ending addresses of the A, X, and Q registers plus the address of AHEXT, a one word extension of the A register, the precision can be changed to any desired length, using 6 bits of precision per register word. For 24 bit precision, four 160 words are used per register, etc. The user is cautioned to make sure that the A-register is positioned such that it is a continuation of the Q-register. For example: in 24 bit precision if the low order address of the Q-register is 7624, the high order address of the Q-register is 7627. In this case the low order address of the A-register must be the next address, 7630, with the high order address at 7633, and AHEXT at 7634. Space has been saved for the registers, beginning at 7620 and ending at 7776, in memory. The following cells of the first sixty-four memory locations hold the seven variable addresses:

AHEXT - 0013 - one word extension of A-register
AH - 0015 - high order address of A-register
AL - 0016 - low order address of A-register
QH - 0024 - high order address of Q-register
QL - 0025 - low order address of Q-register
XH - 0021 - high order address of X-register
XL - 0022 - low order address of X-register

Each data word to be used by the MPPN routine must be packed with the four high order octal digits in the low order address, the next four lower octal digits in the next higher address and so forth. Examples of this packed representation are:

Octal Number (24 bit precision)	Location	Contents
6532 4173	1000	6532
	1001	4173
(42 bit precision)		
56410273561402	2000	5641
	2001	0273
	2002	5614
	2003	0200

Arithmetic is accomplished in 1's complement fashion, with the highest order bit interpreted as a sign bit. For example the 48th bit in 48 bit precision is interpreted as a sign bit following standard conventions of a 1 if negati and a 0 if positive.

1. Principle of Operation - MPPN consists of a control routine and a number of arithmetic subroutines. The control routine makes possible the linkage of any number of subroutines without generating a subroutine linkage in the main program each time one of the MPPN subroutines is use

   The MPPN subroutines are designed to operate using one parameter, the operand address, called PY in the following discussion.

2. Control Routine - The control routine provides the means of tying MPPN subroutines together and performs the function of getting the parameter from memory to a standard location for use by the subroutines.

   The entrance to the control routine is ENTR, or 6300. This entrance causes the control routine to continue to interpret calling sequences until the subroutine BASIC is called.

   A calling sequence consists of the subroutine mnemonic code followed by the parameter, PY, required by the subroutine.

   Example -

   To solve the problem, (A+B) C/D and store the result at E, in MPPN, and return to basic 160 machine coding, the following coding (expressed in Assembly Language) will do the job.

Location	Op	Address	Additive
L1	LDF	03	
L2	JFI	01	
L3		ENTR	
L4		NEXT	
NEXT		LDA	
		A	
		ADD	
		B	
		MPY	
		C	
		DIV	
		D	
		STA	
		E	
		BASIC	
L5			

At conclusion of the store A routine, the control routine recognizes the BASIC, and returns control to the main program at L5.

3. MPPN Subroutines.

a) Load MPPN's A-Register.
Calling sequence
LDA
PY
Operation - Load the octal number beginning at location PY into the MPPN's A-Register.

b) Load Complement MPPN's A-Register.
Calling sequence
LCA
PY
Operation - Load [illegible] complement of the octal number beginning at location PY into the MPPN's [illegible] ster.

c) Store MPPN's A-Register.
Calling sequence
STA
PY
Operation - Store the octal number in MPPN's A-Register beginning at location PY in memory. The 4 high order octal digits go into PY; the next 4 lower order octal digits into PY + 1 etc.

d) Store MPPN's Q-Register
Calling sequence
STQ
PY
Operation - Store the octal number in MPPN's Q-Register beginning at location PY in memory. The 4 high order octal digits go into PY; the next 4 lower order octal digits into PY + 1 etc.

e) Addition
Calling sequence
ADD
PY
Operation - Add the octal number beginning at location PY to the MPPN's A-Register.

f) Subtraction
Calling sequence
SUB
PY
Operation - Subtract the octal number beginning at location PY from the MPPN's A-Register.

g) Multiplication
Calling sequence
MPY
PY
Description - This is a fractional multiply. Therefore the binary point is assumed to be immediately following the sign bit. Example; the fraction 1/2 would be 0.10 - 0 with N bits, the fraction - 1/4 would be 1.101 - 1 with N bits.

Operation - Multiply the octal number in memory location, PY, by the octal number in the MPPN's A-Register. The most significant N bits of the 2N bit signed product are in the MPPN's A-Register with the least significant N bits in the MPPN's Q-Register.

h) Division
Calling sequence
DIV
PY
Description - This is a fractional divide; see description under multiplication. Therefore an overflow condition, with an incorrect answer, will exist if the divisor is smaller, in absolute terms, than the dividend.

Operation - Divide the octal number in the MPPN's A-Register by the octal number, in memory, beginning at location PY. At the conclusion of the divide instruction, the appropriately signed quotient is in the MPPN's A-Register with the signed, same as dividend, remainder in the MPPN's Q-Register. Note that a halt is executed if division by zero is attempted.

i) A Left Shift
Calling sequence
ALS
PY
Description - This is a end around circular left shift of the MPPN's A-Register. In this case, the operand, PY, is a shift count in octal.

Operation - Shift the MPPN's A-Register left end around PY positions.

j) A Right Shift
Calling sequence
ARS
PY
Description - This is an end off right shift of the MPPN's A-Register. In this case, the operand, PY, is a shift count in octal. In this shift operation the sign bit is extended to the right.

F1.

IDENTIFICATION

Title: Paper Tape Verify - Ident. F1.03
Programmer: R. Beale, January 1961

PURPOSE

The program is used to verify copies of an original paper tape.

USAGE

1. Operation procedure:

   a. Master Clear

   b. Machine load Paper Tape Verify program at zero

   c. Master Clear

   d. Insert original tape into reader

   e. Run

   f. On HLT 01, P = 0107; insert second tape into reader and run without altering console

   g. On HLT 02, P = 0130: the last tape is equivalent to original -- to verify another tape, place it in reader and run without altering console -- upon successful verification step g may be repeated.

3. Space Required: $135_8 = 93_{10}$

7. Error Stops: On ERR 01, P = 0132: the last tape is not equivalent to the original -- to continue verifying tapes, place new tape in reader and run without altering console. This will execute Step 1g.

10. Timing: 350 frames/second

13. Equipment Configuration: Minimum System

METHOD OR ALGORITHM

A series of sequence-sensitive check sums is formed for each tape and tested for equivalence. There is no limit on the lengths of tapes to be verified.

FLOW CHART - Not applicable.

F3.

IDENTIFICATION

Title: Listable Octal Dump, Ident. F3.02

Programmer: Don T. Miller, July 1961

PURPOSE

To produce a listable octal dump of a specified area of core.

USAGE

1. Operational Procedure

   The program may be loaded anywhere greater than address 1. The P register should be set to the starting load address, the starting address of the area to be dumped put in the A register, and the Run switch put in RUN. The program will stop at the load address $+12_8$, and the final address of the dump should be placed in A. When put into run mode the program will punch the required area of core (in Flexowriter code), punch a stop code, restore any registers in low core that it uses, and stop at the load address $+105_8$.

3. Space Used: $251_8 = 169_{10}$ locations

9. Output Format: Each line will contain: carriage return, octal address, the octal contents of eight registers, four spaces, octal address, and the octal contents of eight registers. Duplicate lines will not be punched.

10. Timing: The program operates at punch speed.

12. Caution to user: The punched contents of cell 7777 will be the contents of cell 0.

13. Equipment Configuration: Minimum

F3.

IDENTIFICATION

Title: BI-OCTAL DUMP 2, Ident.F3.06
Programmer: H. C. Schnackel modified by C. M. Atchison, April 1961

PURPOSE

This program will sequentially dump in BI-OCTAL (machine-load format), the information stored in core memory beginning with the starting address, and ending at the terminating address minus one. The output tape can be loaded by use of the "Load" reader mode on the 160 computer.

USAGE

1. Operational Procedure

   a. Load BI-OCTAL Program Tape at 7732_8.

   b. Set P register at 7732_8.

   c. Set A register equal to first word location to be dumped.

   d. Run.

   e. Set A register to last word location to be dumped plus one.

   f. Punch on.

   g. Run.

3. Space Required

   a. Decimal -- 37 locations high core

   b. Octal -- 45 locations high core

4. Temporary Storage Requirements

   Uses and restores location 0.

12. Cautions to User

   a. The program is not relocatable.

   b. The program does not punch leader or trailer in the output tape.

II) Remove the cards from the read hopper and reproduce the damaged cards from the bottom of this stack. Replace the reproduced cards on the bottom of the stack and discard the damaged cards.

III) Push the START button until all cards in the machine have been fed out.

IV) Put the last card fed out in front of the reproduced cards from the read hopper.

V) Place the corrected card stack in the read hopper and push the start button twice to feed two cards.

VI) Put the read stacker switch in the OFF position (toggle it away from you). The NOT READY light will go out and the computer will continue reading cards.

2. Parameters

a. Entrance parameter

OWOU as follows:

W = tape control unit number (usually 1)
U = logical tape number

b. Exit parameters

Upon normal return, none.
Upon error return:

I) for bad tape exit, A = 7775.
II) for illegal BCD code, A = column number in which first bad code appears.

3. Space Requirements

1154_8 locations in the relative bank.
20_8 locations in the direct bank.
254_8 locations in the indirect bank.

Locations in the indirect bank are assigned by EQU cards, $124_8 = 84_{10}$ locations for the symbol BLKA and $130_8 = 88_{10}$ locations for the symbol BLKB. This storage assignment must be non-overlapping and location (i) 0000 must not be included.

If it is desired to re-arrange the direct storage assignments, only the symbol LCARD may be assigned location (d) 0000. All other direct storage locations may be assigned at will.

6. Error Return

   a. If the program is unable to write a record 9 times because of parity errors, the error return occurs with A = 7775. In this case, the card which has just passed the 2nd read station has been read but not processed. The card image for the card immediately preceding the above card is the card image which cannot be put on tape.

   b. If an illegal punch combination occurs in a column, the error return occurs with A = first column in which a bad code appears. In this case, the card which has just passed the 2nd read station is in error. The card image for the card immediately preceding the above card is the last card image written on tape.

7. Program Stops

   As a closed subroutine, the only program stop is at symbolic location RR17+1 after sensing an End-of-Tape reflective marker. The program rewind-unloads the tape prior to the stop. Mount the next tape and continue. This stop is at location (r) 1247 when used as an independent program.

   As an independent program, two additional stops are provided: at location (r) 0063 for the normal return and location (r) 0065 for the error return.

   To continue running from any of the above stops, put the RUN-STEP switch into the STEP position, then into the RUN position.

8. Output Tape Mounting

   The output tape is mounted on control unit W with logical selection U as in 2.a. above. The tape must not be file protected.

9. Input and Output Formats

   a. Card Input Formats

      The presence of both 7 and 9 punches in column 1 identifies a column binary card. All other cards are assumed to be BCD.

   b. Tape Output Formats

      1) for binary cards, a 168_{10} character record, the first 160_{10} characters representing the binary card in odd parity bit-for-bit and the next 8_{10} characters as follows:

13. Equipment Configuration

a. Basic 4k 160 with paper tape input and output.

F6.

IDENTIFICATION

Title: 160-A Card to Magnetic Tape Routine - Ident. F6.05
Programmer: Larry Brown, November 1961, Control Data Corporation

PURPOSE

This routine accepts standard 80 column BCD or column binary cards and writes their card images with "look-ahead" bits onto magnetic tape. Programmed for the IBM 533 Card Read Punch.

USAGE

1. Calling Sequence and Operational Procedures

   a. As a closed subroutine, entry is made with the following calling sequence:

```
L      JPR
L+1    CDTAP
L+2    NORMAL
L+3       RETURN
L+4    ERROR
L+5       RETURN
L+6    OWOU
```

   b. As an independent program, load the bi-octal tape at location 0000 in either bank, and start at location (r) 0060 with OWOU in the A-register.

   c. 533 Card Read Operation

      Place the cards to be read in the read hopper to feed 9-edge first, face down. Push the START button twice to feed two cards and turn off the NOT READY light. At least two blank cards must follow the deck to be read.

   d. Card Feed Failure

      In case of a card feed failure, follow these procedures:

      I) Put the read stacker switch in the <u>ON</u> position (toggle it toward you).

00 05 00 01 00 05 00 04 if the next image on tape is a binary image in odd parity,

00 04 00 00 00 01 00 00 if the next image on tape is a BCD image in even parity.

II) for BCD cards, a 88_{10} character record, the first 80_{10} characters in standard BCD representing the BCD card in even parity and the next 8_{10} characters as follows:

11 11 07 07 20 20 20 20 if the next image on tape is a binary image in odd parity,

20 20 20 20 20 20 20 20 if the next image on tape is a BCD image in even parity.

c. Control Card Formats

I) FILE* in columns 1-5 causes an End-of-File mark to be written on tape. The FILE* card image is not written on tape. The program continues with the next card image following the EOF mark.

II) ENDbb in columns 10-14 causes an End-of-File mark to be written on tape following the END card image. The program continues with the next card image following the EOF mark.

III) FINIS in columns 10-14 causes two End-of-File marks to be written on tape following the FINIS card image. The program rewind-unloads the tape and exits to the normal return. Use of the FINIS card is the only way to legitimately terminate the operation of this program.

10. Timing

Approximately 200 cards per minute read and put on tape. This program operates as fast as the 533 card read unit.

12. Cautions

Do not step through the routine or take the computer out of RUN, else the input-output transfers may not be completed correctly.

13. Equipment Configuration

Minimum 160-A with 163 and 1610 adaptor for the IBM 533 card read punch. The program assumes a 2-bank 160-A. The 163 tape unit is selected in both the character (6-bit) and assembly (12-bit) modes. The buffer channel is used to read cards. Cards are read from the second read station.

15. Miscellaneous

a. Normal Return

Follows reading a FINIS card.

b. Interrupt

Not used.

c. Bank settings

Upon normal and error exits, the direct, indirect, and buffer bank assignments are restored to their values at time of entry to the subroutine. The subroutine will work in either bank.

d. Machine Registers

The Buffer Entrance, Buffer Exit, and A-register are not restored.

e. Legal BCD codes and card column punch codes as per standard set.

f. Selective Stop and Selective Jump switches are not used.

METHOD

1. Tape write error procedures.

Following detection of parity errors on writing the tape, a backspace and a new write operation is performed. If 3 tries at writing fail, the tape is backspaced, an EOF written, and a backspace is performed to skip over about 6 inches of tape and 3 more attempts are made to write the record before another EOF-backspace is tried. After 9 attempted write operations fail, the tape is backspaced and the error return is made with A = 7775.

2. Checking for illegal BCD codes is accomplished by the following equations operated on each 12-bit word of the card image read in. The * indicates a bitwise logical product and the + indicates a bitwise logical (exclusive) OR. The numbers 1, 2, 3.... represent the card row 1, 2, 3....etc. If the following equations are not satisfied, an illegal BCD code appears on the card.

7*9=0
(7+9)*6=0
(6+7+9)*5=0
(5+6+7+9)*2=0
(2+5+6+7+9)*1=0

```
(1+2+5+6+7+9)*8=0
 3*4=0
(1+2+5+6+7+9)*(3+4)=0
(1+2+3+4+5+6+7+8+9)+8*(3+4)  *0*11=0
(1+2+3+4+5+6+7+8+9)+8*(3+4)  *0*12=0
 11*12=0
```

APPENDIX

References

160 FORTRAN/Reference Manual, Control Data Corporation, Minneapolis, Minnesota, April 1962.

AUTOCOMM/General Information Manual, Control Data Corporation, Minneapolis, Minnesota.

INTERFOR/Reference Manual, Control Data Corporation, Minneapolis, Minnesota.

OSAS-A/The 160-A Assembly System, Control Data Corporation, Minneapolis, Minn.

Programmer's Reference Manual for Four-Bank INTERFOR Programming System, Defense Research Laboratories, Santa Barbara, California, March 1962.

SICOM - Flow Diagrams and Coding Sheets, Scientific Computers, Inc., Minneapolis, Minnesota, June 1962.

SICOM - Scientific and Commercial Programming System for Control Data 160-A Computer, Scientific Computers, Inc., Minneapolis, Minnesota, June 1962.

DISTRIBUTION LIST

Bennett, R. P.	14059	(50 Copies)
Biggar, D.	24090B	(2 Copies)
Busch, R. E.	24065B	(2 Copies)
Champaign, M. E.	24127B	(2 Copies)
Ciaccia, B. G.	24082	(2 Copies)
Derango, W. C.	24082B	(2 Copies)
Greenwald, I. D.	24058A	
Henley, D. E.	24058B	
Kneemeyer, J. A.	24065A	(2 Copies)
Kostiner, M. N.	14056	(20 Copies)
Marioni, J. D.	24076B	
Munson, J. B.	24048A	
Olin, O. C.	14054	
Polk, T. W.	24099	
Rosenberg, E. J.	14050	(3 Copies)
Skelton, R. H.	24127	(2 Copies)
Stone, E. S.	22116A	
Tennant, T. C.	27024	
Totschek, R. A.	24090A	(2 Copies)
Vorhaus, A. H.	24076A	(2 Copies)
West, G. P.	24094A	
Wong, J. P.	Sunnyvale	(15 Copies)

PIR-E5 Aerospace

R. E. Olsen
T. R. Parkin

UNCLASSIFIED

System Development Corporation,
Santa Monica, California
160-A REFERENCE MANUAL.
Scientific rept., TM(L)-993, by
E. J. Rosenberg. 8 February 1963,
281p., 7 refs.
(Contract AF 19(628)-1648, Space
Systems Division Program, for Space
Systems Division, AFSC)

Unclassified report

DESCRIPTORS: Programming (Computers).

Describes and lists all 160-A computer
routines available in the Computer

UNCLASSIFIED

Program Development Center (CPDC).
Serves as a training aid for new
CPDC personnel.

UNCLASSIFIED

UNCLASSIFIED