

6000 SYSTEMS DIVISION

CONTROL DATA®

SCOPE/INTERCOM

GENERAL INFORMATION MANUAL

CONTROL DATA
CORPORATION

PREFACE

The 6000 Computer Systems Division of Control Data Corporation has developed the Control Data SCOPE/INTERCOM. This software package provides an on-line time-sharing capability for users of Control Data 6000 Series computers. The system permits remote access of the central computer resources by a number of simultaneous users. The time-sharing system operates concurrently with the conventional batch mode in process at the central site.

The SCOPE/INTERCOM software package is available on an individual quotation basis through the 6000 Computer Systems Division.

This manual assumes familiarity with Control Data 6000 Series computer system hardware and software. For additional information concerning the 6000 Series computer systems and the SCOPE/INTERCOM interface, refer to the following manuals:

<u>Title</u>	<u>Publication Number</u>
Control Data 6400/6500/6600 Computer Systems Reference Manual	60100000 Rev. B
Control Data 6400/6500/6600 SCOPE Reference Manual	60189400

CONTENTS

1 . .SCOPE/INTERCOM DEVELOPMENT	1-1	Peripheral Processor Programs	3-3
Introduction	1-1	Files	3-3
Purpose	1-1	Utilities	3-4
Advantages	1-1		
2 . .HARDWARE CONFIGURATION	2-1	4 . .SYSTEM-OPERATOR COMMUNICATION	4-1
3 . .SOFTWARE ORGANIZATION	3-1	Central Site Operation	4-1
Security	3-1	Remote Terminal Operation	4-1
Central Memory and other Requirements	3-1	Terminal Interaction	4-2
Pointers	3-1	Supervisory Edit and Transfer Utility Program	4-2
Channels	3-1	Introduction	4-2
File Name Table/ File Status Table	3-1	Organization	4-2
User Table	3-2	Editing	4-3
Password Table	3-2	5 . .BASIC LANGUAGE	5-1
Buffer Tables	3-2	General	5-1
Control Point Area	3-2	Description	5-1
		Statement Format	5-1

FIGURES

2-1 SCOPE/INTERCOM System Configuration	2-2
---------------------------------------------------	-----

INTRODUCTION

SCOPE/INTERCOM evolved from the Sharer System which was developed as a time-sharing system for the Control Data 6600 Computer with a 131K memory. The original Sharer System ran concurrently with batch processing and required 50K memory locations. Its general purpose was to provide from teletypes: file input, editing, manipulation, compilation, and interacting execution.

Substantial modifications to the system's internal operation enable the new system (SCOPE/INTERCOM) to be used on a machine with a 65K memory.

PURPOSE

This document describes the general features of SCOPE/INTERCOM. The purpose of the software capability is to direct a flow of information and data between a 6000 computer and a number of remote TTY and CRT stations. A number of users will be simultaneously provided with the on-line capability to:

- Create programs
- Define, store, reference, and edit files
- Submit newly created programs to be run at the SCOPE/INTERCOM control point or submit them to the batch input queue to be run
- Submit FORTRAN and BASIC programs at the SCOPE/INTERCOM control point and interact with the executing programs from the remote terminal
- Request that output from jobs run in the batch queue be saved as a common file for later retrieval by the remote user
- Request input or output through devices at the central site.

These capabilities will co-exist with the 6000 computer batch processing capabilities.

ADVANTAGES

SCOPE/INTERCOM has two principal advantages over batch processing:

1. SCOPE/INTERCOM offers faster turn around time than that offered by conventional batch processing.

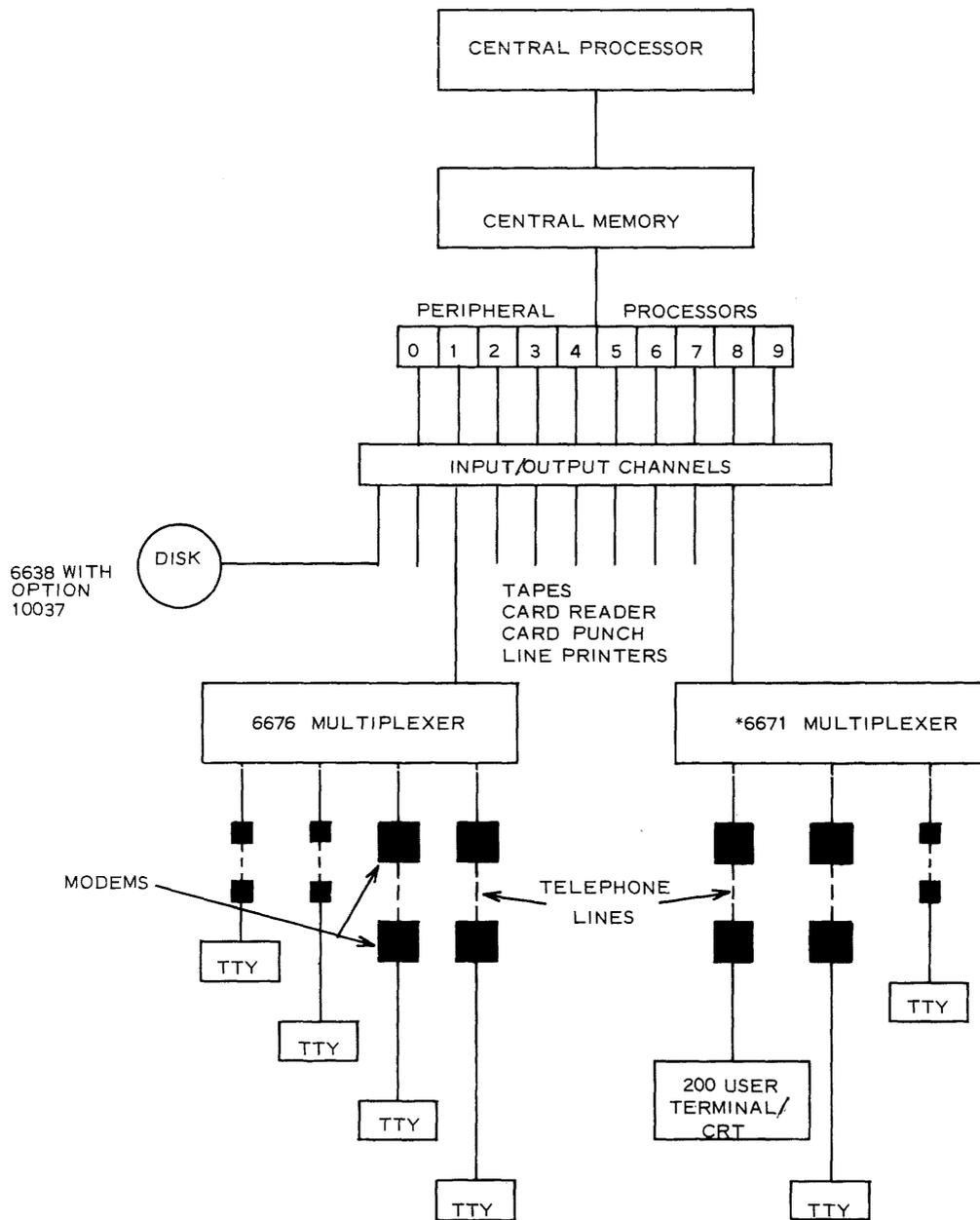
2. SCOPE/INTERCOM permits solution of problems requiring human judgement to direct a process toward an acceptable solution.

For example, problems arising in on-line programming are rapidly solved as SCOPE/INTERCOM permits the programmer to construct programs, test them, make corrections, and execute them without the inefficiencies and delays of batch submittal. Also source language program files can be created and edited on-line by the programmer, the file submitted to a compiler for translation, and the resulting object program executed for immediate evaluation by the programmer.

In addition, interactive problem solution allows managers, accountants, engineers, researchers, students, etc. to observe their problem solution in progress, make decisions and adjustments, and direct the process to an acceptable solution, thus avoiding the delay and expense of repeated batch executions.

Figure 2-1 depicts a typical SCOPE/INTERCOM equipment configuration. This configuration includes all of the equipment necessary for the operation of a SCOPE Operating System. It is necessary to have a 6671 or 6676 multiplexer. Multiple multiplexers will be permitted with an accompanying requirement for more dedicated PPU's and dedicated channels. The 6671 can accommodate up to 16 TTY's or 16 200 User Terminals or a mixture of these two devices. The 6676 can accommodate up to 64 teletypes. The SCOPE/INTERCOM system will handle up to 64 teletype remote terminals. While response time is dependent on a number of factors such as the types of jobs, job size and frequency in the number of disk accesses, it will also increase as the number of active terminals increases. Models 33 and 35 teletypes will be supported.

If RESPOND or EXPORT/IMPORT 200 are to be run at the same time as SCOPE/INTERCOM, they will require a separate control point and a separate multiplexer.



*EXPORT/IMPORT 200 USER TERMINAL FOR REMOTE BATCH REQUIRES SEPARATE MULTIPLEXER

Figure 2-1. SCOPE/INTERCOM System Configuration

SECURITY

In order to gain access to the SCOPE/INTERCOM system, a user must "LOGIN" and supply a user name and an associated password. Each user name/password combination in the SCOPE/INTERCOM system may have restrictions on the use of the system. A user may have restrictions on his field length, maximum time allotment, and the number of files that he may save. Each user is assigned an access level which will determine which programs in the system he is permitted to use. Files created by one user are protected from other users (see SCOPE/INTERCOM files, page 3-3).

CENTRAL MEMORY AND OTHER REQUIREMENTS

SCOPE/INTERCOM will require approximately 1000_{10} central memory words for PPU programs resident in central memory resident; 25 words for each TTY in the system; 10 words for each backup buffer; and six words for each recognizable password in the password table. SCOPE/INTERCOM also requires a dedicated multiplexer, data channel, and PPU to service the remote terminals. SCOPE/INTERCOM can run with as little as one control point or as many as seven control points. The number of control points assigned to SCOPE/INTERCOM is an operator directive. A dedicated access of a mass-storage device is required to ensure adequate response time.

POINTERS

SCOPE/INTERCOM will use three pointers in central memory resident. The first pointer will be the SCOPE/INTERCOM queue pointer which will give current information on the status of the job swap queue. The second and third pointers will give information about SCOPE/INTERCOM tables located in central memory.

CHANNELS

Two pseudo channels are used for SCOPE/INTERCOM. The first is used to lock the job swap queue while it is being altered. The second is used while information is being transferred to the TTY I/O Buffers.

FILE NAME TABLE/FILE STATUS TABLE

There will be an entry created in the File Name Table (FNT) for each user when he enters the system by the command "LOGIN". This entry is called the Master File Name Table

(MFNTE) for that user and it will contain the disk address that is used for swapping the job in or out, pointers to other user MFNTE entries, and pointers to other files of this user. The entire chain of MFNTE entries is called the SCOPE/INTERCOM queue.

USER TABLE

There is a user table for each possible remote terminal in the system. It will contain information concerning maximum field length, time limit, access level, number of files, time used, etc., for each remote terminal that is "logged in."

PASSWORD TABLE

There will be a password entry for each possible password in the system. The user may elect to incorporate special modifications to restrict his own authorization file.

BUFFER TABLES

There will be three types of buffer tables: input buffer, output buffer, and a backup buffer. Input buffers will be used to hold messages originating at the remote terminal. Output buffers will be used to hold messages to be output to the remote terminal. Backup buffers can be used to hold overflow messages from the output buffers. There must be an input and output buffer for each remote terminal in the system. The number of backup buffers will be an installation parameter.

CONTROL POINT AREA

SCOPE/INTERCOM will use three locations in the control point area for pointers and for the recall of PPU programs.

CONTROL POINTS

The number of control points reserved for SCOPE/INTERCOM or for batch activity is an operator decision. When the operator types N.INTERCOM, control point N is reserved for the running of SCOPE/INTERCOM jobs. When the operator types N.NEXT, control point N is reserved for the running of batch jobs. Typing N.DROP at a SCOPE/INTERCOM control point will remove the reservation of that control point but the job currently running at that control point will be returned to the SCOPE/INTERCOM queue and eventually run at another SCOPE/INTERCOM control point.

There is no resident core at the SCOPE/INTERCOM control point that is needed for the operation of SCOPE/INTERCOM. All of the SCOPE/INTERCOM operations will be performed by transient peripheral processor programs. The only memory required at a control point will be that amount of core necessary to run the user's jobs.

PERIPHERAL PROCESSOR PROGRAMS

There are five PPU programs that currently make up the SCOPE/INTERCOM system:

- INT -Called when the operator types N.INTERCOM at a control point. INT reserves the control point for the SCOPE/INTERCOM system, initiates 1BR and the multiplexer drivers if they are not already initiated, and calls 1SJ to service this control point.
- 1BR -Transient PPU program residing at control point 0, which searches inactive remote terminal buffers for "LOGIN " commands, services the backup buffers, and oversees control point activity.
- 1SJ -Transient PPU program residing at a SCOPE/INTERCOM control point which is responsible for the swapping of jobs and the maintenance of the SCOPE/INTERCOM queue. Programs will be swapped only if there are other programs on mass storage devices that need processing. Swapping will occur after a preset time interval or sooner. If a transfer of data between the central program and the remote terminal is required but cannot occur because of terminal activity or user inactivity, the job may be swapped out.
- M71 -Multiplexer driver for the 6671 multiplexers.
- M76 -Multiplexer driver for the 6676 multiplexers.

FILES

Under SCOPE/INTERCOM, three types of files exist:

1. User private files
2. Common files
3. System files

User private files are created by an individual user and can be read, altered, or deleted only by the originator.

Common files are user private files that have been marked by the originator as being accessible to other users. Other users may read, alter or delete them.

System files are files that exist in the SCOPE library. These files can be loaded by any user but cannot be modified or deleted by any user.

User files created under SCOPE/INTERCOM can be retained indefinitely through use of an "archives tape" created by SCOPE/INTERCOM upon an operator command at the end of the "system day." All user private files marked for saving are written on a magnetic tape at

this time. At the start of the next "system day," the magnetic tape is re-read upon operator command and the user files are restored to their condition at the end of the previous day.

When a "permanent files" capability is provided under the SCOPE Operating System, that feature will be used for user file retention.

User private files are protected under SCOPE/INTERCOM and they cannot be read or altered by anyone but the originator. If it is desired to make the contents of a private file available to other users, a duplicate of the private file can be written and the duplicate made a common file. This common file is available to all users while the original remains secure.

UTILITIES

A SCOPE/INTERCOM user may access any of the standard SCOPE library routines (except EDITLIB) to help in the solution of his problem. In addition, SCOPE/INTERCOM will provide many other standard programs to assist the user. These utility programs will enable the user to create, edit, sequence or erase his own private files or submit these files for batch execution, printing or punching at the central site. A BASIC compiler will also be provided under the SCOPE/INTERCOM system enabling a person with little or no programming background to make use of the 6000 computer.

Descriptions of the Supervisory Edit and Transfer Utility Program (SETUP) and the BASIC compiler are provided as separate sections of this document (see sections 4 and 5, respectively).

CENTRAL SITE OPERATION

SCOPE/INTERCOM operation is initiated by the central site computer operator placing the pseudo-job "INTERCOM" at one or more SCOPE control points. When the first control point is so named, the resident peripheral processor program for driving the 6676 or 6671 multiplexer(s) is loaded and started. As user commands are received from terminals, the command processing programs are called from mass storage to a SCOPE/INTERCOM control point for execution. When there are more user programs active than there are control points assigned to SCOPE/INTERCOM, the control points are shared among the programs through periodic program swapping. The field length assigned to a SCOPE/INTERCOM control point is automatically chosen to suit the requirements of the user program. When a SCOPE/INTERCOM control point is not in use, its field length is reduced to zero, thus releasing memory space for other SCOPE/INTERCOM control points or for batch jobs. Central processor time is apportioned as usual by SCOPE to jobs running at SCOPE/INTERCOM control points and to jobs running at other control points, with two exceptions:

1. SCOPE/INTERCOM programs are given a higher priority than batch jobs, giving SCOPE/INTERCOM programs preferential treatment over lower priority batch jobs.
2. If insufficient memory space exists to read in a user program at a SCOPE/INTERCOM control point, one or more lower priority batch jobs may be automatically rolled -out to make room.

The computer operator can allocate computer resources between SCOPE/INTERCOM and other programs by assigning an appropriate number of control points to SCOPE/INTERCOM. Thus, the varying requirements of SCOPE/INTERCOM users and batch load can be balanced.

REMOTE TERMINAL OPERATION

To initiate operation under SCOPE/INTERCOM, a user logs in at a remote terminal and enters a user name and password. Upon user log-in, SCOPE/INTERCOM verifies the name and password and assigns the user his allotment of memory space and computer time and his access level. The user then uses the SCOPE/INTERCOM facilities as desired within the above restrictions. Operation with SCOPE/INTERCOM is terminated by the user when he logs out at the terminal. SCOPE/INTERCOM computes total central processor and peripheral processor time expended by the user during his period of SCOPE/INTERCOM operation.

TERMINAL INTERACTION

A user at a remote terminal can request any SCOPE program (except EDITLIB) or any SCOPE/INTERCOM utility program to be loaded at a SCOPE/INTERCOM control point. Before this program is loaded, a check is made on the user field length and the access level to determine if the user is allowed to use this program. Assuming proper authorization, the program is loaded and executed. During the execution of FORTRAN (RUN #2.3) and BASIC programs, interaction may take place between the executing program and the remote terminal.

In FORTRAN, it is necessary for the user to call a system subroutine (called TYPEIT) at the beginning of his program. This subroutine will link his READ and PRINT statements to the remote terminal.

In BASIC, INPUT and PRINT statements will be linked automatically to the remote terminal.

SUPERVISORY EDIT AND TRANSFER UTILITY PROGRAM

INTRODUCTION

Inherent in the SCOPE/INTERCOM system is a general purpose utility (Supervisory Edit and Transfer Utility Program[SETUP]), which is designed to provide the user with some degree of flexibility at the remote terminal in building and running programs. SETUP enables one to create or modify a source program without the restrictions of precise card format, and subsequently to submit the program for compilation and execution. While SETUP has its own commands, all SCOPE/INTERCOM commands are accessible through it.

ORGANIZATION

When SETUP is called, the user is supplied with a text-buffer whose length can be increased, within limits, in the event more space is needed. Into this buffer the user enters a program either already coded and stored on disk, or one which is being constructed line-by-line at the remote terminal. At any time, he may edit or otherwise manipulate the program to suit his needs; e.g., eliminate known bugs, correct misspellings. When the user is satisfied with the final coding, he may submit it to the appropriate compiler and, if there were no compile errors, execute the object program. It should be noted here that in the event the compiler does produce error diagnostics, the text buffer will not have been destroyed and thus can be edited further and resubmitted. The contents of the text buffer remain intact unless altered by the user, being erased only when the user logs out. (Note: If the user wishes to use the file in the text buffer at some later time, he must remember to store the text buffer with a SAVE command before he logs out.)

2. What is the number of the line I want to change?

A line is changed by retyping the entire line.

To insert a new line, type the SETUP line number followed by the statement desired.

If this line number is already in the text buffer, that line will be replaced by the line just typed.

If this line number is not already in the text buffer, the line just typed will be inserted into the text at the appropriate sequential place.

If a line number is typed with no accompanying characters, the corresponding line in the text buffer is deleted.

Several keys of the remote terminal have special editing significance with respect to the physical line being typed. For example, on a teletype, the carriage return key erases that entry line. Typing (!) deletes the preceding character. Thus

```
X !GEE !ORY !GE WASING !!!HINT !GTON
```

was our first president.

The slash (/) has a special meaning in SCOPE/INTERCOM. If one is aware of which questions will be asked by the computer, they can be anticipated and time saved by use of the slash.

For example:

```
UNSAVE/FILNAME
```

avoids a question asking the file name after the UNSAVE directive.

The following is a list of the directives:

<u>DIRECTIVE</u>	<u>PURPOSE</u>
BASICX	Calls the BASIC compiler to compile the lines of the text buffer as a program. If there are no errors discovered by the compiler, the program will be executed. If there are errors, an error listing will be presented on the remote terminal.
BYE	Returns the user to the SCOPE/INTERCOM COMMAND Mode.
DELETE	Has two formats which delete lines from the text buffer.
DELETE X	Deletes line number X.
DELETE/X/Y	Deletes all lines from X to Y.

DIRECTIVE

PURPOSE

GOODBYE	Same as BYE.
LENGTH	Prints the number of lines currently in the text buffer, and the current field length of SETUP.
LIMIT	When a program is executed from SETUP it is assigned a time limit. LIMIT will augment that value if needed.
LIST	Prints out the text buffer.
LIST/X	Prints the line of the text buffer numbered X.
LIST/X/Y	Prints the text buffer lines X to Y. (X and Y can be from one to five characters.)
NEW	Indicates that a new file is about to be defined.
OLD	Indicates that an already defined file is desired.
RENAME	Allows one to change the name of the current file in the text buffer.
RESEQ	Resequence the line number for each line in the text buffer. The initial line is assigned the value 100, and successive lines are incremented by 10 from that point.
RUNX	Calls the FORTRAN compiler to compile the lines of the text buffer as a program. If there are no errors discovered by the compiler, the program will be executed. If there are errors, an error listing will be presented on the remote terminal.
SAVE	Saves, on disk, the current text buffer under its file name.
SCRATCH	Erases the text buffer.
SYSTEM	Allows the selection of the mode being used. This can be FORTRAN, BASIC, or GENERAL.
UNSAVE	Removes a file from disk storage, and erases it.

This section briefly describes the BASIC programming language to be provided under the SCOPE/INTERCOM time sharing system.

The general characteristics of the language are discussed, specific formats and capabilities described, and an example of a BASIC program given.

GENERAL

The simplicity of the BASIC language makes it especially suitable for use by persons with little or no programming background. The language is intended primarily to be used for mathematical computation, and can be used for a wide range of problem complexity from the very simplest to very complex.

The fundamentals of the BASIC language can be learned very quickly by a new user; then, as the user's proficiency increases, he can use the more complex, but powerful, features of the language.

The format of statements in BASIC is simple and unrestrictive. Errors are easily corrected through inserting, changing, or deleting statements.

Conversational interaction between user and program is facilitated by the availability of BASIC commands for reading from /or printing on the teletype.

DESCRIPTION

Programs written in the BASIC language consist of a series of instructions to the computer. Each statement describes one step in the solution of the problem being solved. The user types the statements as successive lines on the teletype. As the user types, characters are simultaneously printed on the teletype and sent to the computer. When a complete program has been entered, the user types a command to execute the program.

STATEMENT FORMAT

Each BASIC statement contains a line number and one or more BASIC formulas. The line number is an integer from one to five digits and identifies the statement for two purposes:

1. Before the program is executed, the BASIC compiler will order the program statements according to the line number.

2. Line numbers identify the statement to receive control during transfer-of-control operations.

A BASIC formula is an instruction to the computer to perform an operation. Formulas are composed of operators, variables, numbers, functions and subroutines. These are grouped together by parentheses to form expressions.

OPERATORS

Operators are symbols used to express a relation between numbers, variables, functions or other expressions.

The operators are as follows:

- Arithmetic Operators
 - + add
 - subtract
 - * multiply
 - / divide
 - ↑ raise to a power
- Relational Operators:
 - = equal to
 - <> not equal to
 - < less than
 - <= less than or equal to
 - > greater than
 - >= greater than or equal to
- Replacement Operator:
 - = make equal to

With these operators, the user can perform all common arithmetic operations, and test for a relationship between two quantities.

The order in which computations indicated by arithmetic operators are performed depends on parentheses, operators used, and order as follows:

1. An expression enclosed by parentheses will be evaluated before the expression itself is used in a computation.
2. Within an expression, power operations are performed first, multiplication and division next, and addition and subtraction last.

3. Within the constraints of 1 and 2, computation proceeds from left to right.

The order within compound relational operators (<>, >=, <=) is not significant; that is <=, for example, is the same as =<.

FUNCTIONS

The user can evaluate several mathematical functions through use of the following:

- SIN (expression) - find sine of expression
- COS (expression) - find cosine of expression
- TAN (expression) - find tangent of expression
- ANT (expression) - find arctangent of expression
- EXP (expression) - find $e^{(expression)}$
- LOG (expression) - find natural logarithm of expression
- ABS (expression) - find absolute value of expression
- SQR (expression) - find square root of expression
- INT (expression) - find greatest integer not greater than expression.
- RND (expression) - produce random number

Angles in trigonometric functions are expressed in radians.

NUMBERS

Numbers used in BASIC expressions may be integers, mixed numbers, or of the form:

vvvvEp

where vvvv is an integer or mixed number, and p is a signed power of ten. For example, 14,272 could be written 14.272E3. Numbers must be within the range 10^{-308} to 10^{337} .

VARIABLES

Variables in BASIC represent numbers whose value is not initially known. Variables are named with a single letter followed by an optional single digit. Variables are assigned values by LET, READ, or INPUT statements.

ARRAYS

A multi-dimensional unknown may be represented by an array. An array is named with a single letter followed by an optional single digit. The array name is followed by up to three subscripts which define the portion of the array to be treated in the statement in which the array name appears.

BASIC STATEMENTS

The following mnemonics provide the means of describing steps in the solution of a problem.

<u>Mnemonic</u>	<u>Meaning</u>	<u>Example</u>
LET	Perform computation and assign answer to named variable.	LET A=B+X/2
GO TO	Transfer control to another statement.	GO TO 30
IF-THEN	Test relationship.	IF A=B THEN....
DIM	Reserve space for an array with a dimension greater than 10.	DIM A2 3, 12, 10
FOR, NEXT	Create a "loop" for repetitive operations.	FOR X=1 TO 6 NEXT X
END	Indicate end of program	END
STOP	Indicate termination of program operation other than at physical end of program.	... STOP END
REM	Insert explanatory note or remark.	REM X NOW HOLDS SQUARE ROOT
DATA	Supply data to be retrieved by READ command.	DATA 1, 6, -3, 7
READ	Assign named variable(s) values from DATA statement.	READ X, Y, Z
RESTORE	Restore place-pointer in list of DATA quantities to beginning of list.	RESTORE
PRINT	Print message or variable(s) on teletype.	PRINT "ANSWER IS" PRINT X, Y
INPUT	Read data from teletype and assign to named variable(s). BASIC prints question mark as a sign for the user to enter data.	INPUT X, Y, Z
GOSUB	Transfer control to subroutine. Control will return to next statement after GOSUB.	GOSUB 45
RETURN	Return control from subroutine to statement following GOSUB.	RETURN

<u>Mnemonic</u>	<u>Meaning</u>	<u>Example</u>
DEF	Define a frequently used function.	DEF FNX(X) = X/SIN(X)

The following mnemonics are used to manipulate matrices:

MAT READ	Assign named array(s) values from DATA statement.	MAT READ A, B, C
MAT PRINT	Print named array(s) on teletype.	MAT PRINT A, B, C
MAT - -	Matrix manipulation.	
	Add two matrices.	MAT A=B+C
	Subtract two matrices	MAT A=B-C
	Multiply two matrices.	MAT A=B*C
	Invert the named matrix.	MAT A=INV(B)
	Transpose the named matrix.	MAT A=TRN(B)
	Multiply the named matrix by expression.	MAT A=(expression)*B
	Fill out the named matrix with zeros.	MAT A=ZER
	Fill out the named matrix with ones.	MAT A=CON
	Make the named matrix an identity matrix.	MAT A=IDN

ERROR MESSAGES

Various error messages can be produced by BASIC. The messages are printed on the teletype. Messages may describe errors discovered in the program descriptions, e.g., violation of statement rules, unintelligible commands, etc. Errors may also describe a condition which arises during the execution of a program, such as attempted division by zero or a read statement when data is not available.

COMMANDS

The following example illustrates the use of several BASIC commands.

The example reads the cubes of the numbers one to fifteen into array C, then accepts requests from the operator for the printout of any of these cubes. If the operator requests the cube of a number greater than fifteen, the program stops.

<u>Program</u>	<u>Explanation</u>
10 DIM C(15)	Reserve 15 locations for array C.
20 FOR X=1 TO 15	Initialize loop to load array with cubes.
30 READ C(X)	Assign one element of array C the value of the next quantity from DATA statements.

<u>Program</u>	<u>Explanation</u>
40 NEXT X	Repeat 30 with X+1 if X not equal to 15.
50 PRINT (CUBES AVAILABLE)	Print CUBES AVAILABLE on teletype.
60 INPUT X	Prints question mark on teletype; when user types a number, set X equal to number.
70 IF X < 16 THEN GO TO 100	Transfer control to statement number 100 if number typed in is less than 16.
80 REM NUMBER TOO BIG	Remark
90 STOP	Stop the program.
100 LET Y=C(X)	Retrieve cube of type-in number from array C.
110 PRINT X, Y	Print number and its cube.
120 GO TO 60	Transfer control to statement number 60 to get next number from user.
130 DATA 1, 8, 27, 64, 125, 216	Cubes of numbers 1-6.
140 DATA 343, 512, 729, 1000	Cubes of numbers 7-10.
150 DATA 1331, 1728, 2197, 2744, 3375	Cubes of numbers 11-15.
160 END	End of program.