Argonne Aational Laboratory AN INTRODUCTION TO 704 FORTRAN by

G. S. Pawlicki



LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

ANL-6542
Mathematics and
Computers
(TID-4500, 17th Ed.)
AEC Research and
Development Report

ARGONNE NATIONAL LABORATORY 9700 South Cass Avenue Argonne, Illinois

AN INTRODUCTION TO 704 FORTRAN

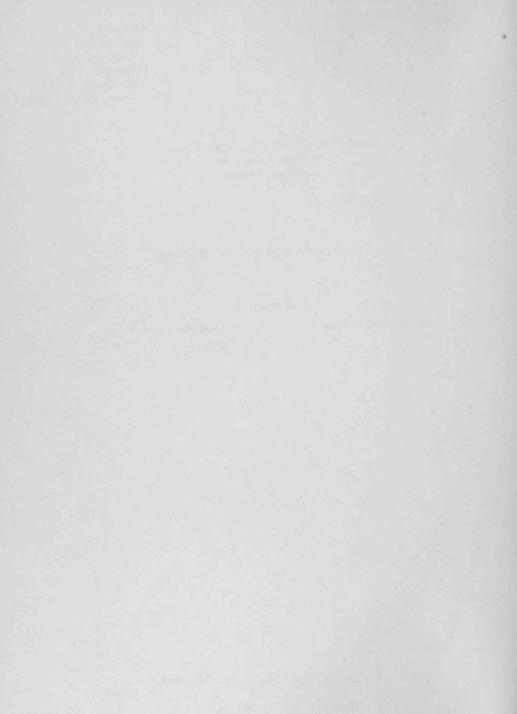
by ____

G. S. Pawlicki

International Institute of Nuclear Science and Engineering

March 1962

Operated by The University of Chicago under Contract W-31-109-eng-38



AN INTRODUCTION TO 704 FORTRAN

by

G. S. Pawlicki

ABSTRACT

It is hoped that this paper will permit the reader to begin to program computations on the 704 computer without referring to the Fortran Manual. Only a fraction of the possibilities of Fortran are described, but at all times an effort has been made to indicate where only a partial expose has been made. Many specific examples are included to illustrate the word descriptions of Fortran programming. No information is supplied concerning operation of the 704 control console and accessories. These functions are performed by personnel of the Applied Mathematics Division. The main body of the paper is devoted to Fortran programming, and an appendix, which gives specific information concerning procedures of the Applied Mathematics Division relative to the use of the 704, is included.

FORTRAN is an abbreviation of the name "Formula Translator." It is called a formula translator, because formulas written almost like ordinary algebra are read from a hand-punched set of IBM cards by the 704 acting as a translator, and a set of cards is punched which serve to operate the 704 as a computer. The set of hand-punched cards is called the "Source Deck." The source deck is translated by the computer, which punches a translated deck of cards called the "Object Deck." The process of preparing an object deck from the source deck is called "Compiling." At "Object Time," that is, when a problem is being solved, the object deck is placed in the card reader, possibly followed by hand-punched "Data Cards" which supply numerical data required by the program. Data can also be read from magnetic tape, but this will not be considered.

The IBM cards used in the source deck and for data have a row of printed numbers from 1 to 80 which identify the 80 columns into which rectangular holes can be punched. The holes, which are punched in a single column, are coded to represent a unique character. The characters are the symbols - +, () *. / as well as numerals and capital letters. At the top of each column is a space in which the card-punching machine also prints the character corresponding to what has been punched. Columns 73 to 80 inclusive are not read by the computer and can therefore be punched with identification such as numerical sequence of source cards and identification of cards as data and their sequence.

AN INTERPRETATION TO THE PORTILAR

Cos. Paul Man

LOAHIBLA

Death to provide and contratable of the contratable

LONTRAM is an abirevisitor of the raine Niconals Translator of the raine Niconals Translator of the residue at the control of the residue at the control of the control of

refer to expend the continue of the source decided to compiling." At Copies to the continue of the object decices along the decided the source of the continue of the source of the sour

The 12M territory of may 1 to 50 equipmentally the 50 columns the velicity of the formal representation of the following state of the fol

To describe the use of columns 1 through 72, probably it is best first to discuss the form of numbers which can be entered as data. Numbers can be entered as floating-point decimal numbers in normal or exponential representation, integers, and octal numbers. Only decimal and integer representation will now be considered. The normal decimal representation of a number will henceforth be referred to as "Floating Point." An integer represented without a decimal point will be called "Fixed Point." Any number is represented by punched holes in a certain series of columns on the IBM cards. The number of columns assigned to a specific number is called the Field Width.

The 72 columns available for numerical input of data can be divided into any desired number of fields of any width desired. The source deck must contain a FORMAT statement which instructs the computer in which way it should interpret the data card. The specification in the FORMAT statement for the floating-point format is

n F w · d

where n is the number of identical fields to be read (the first field begins with column 1!); F is the code letter for floating-point representation; w is the field width; and d is the number of places to the right of the decimal point.

The specification of a fixed-point number in a FORMAT statement is of the form

nIw

where n and w are as previously defined and I is the code letter for fixed-point representation. In punching a data card with a number, the sign precedes the number. A floating-point number can be punched anywhere within the field. With a fixed-point number the number must be at the right side of the field; otherwise, the blank spaces at the right end of the field, read by the computer as zeros, would alter the number by powers of ten. In both representations, if the numerical sign is missing, the computer assumes that the number is positive.

Only single-line format for numerical quantities will be described. The source card for a FORMAT statement might be punched as follows:

Column 1 Blank

Columns 2 - 5 A Statement Number located anywhere

in this field, an integral number with-

out sign.

Column 6 Blank

Column 7 onward to 72 might be punched as follows:

FORMAT (3 F 6.2, F 5.4, I2, 3I4).

To discuse the form of number's which can be entered as data. Numbers can be entered as form, should be entered as form, should be entered as flowing point decimal amorbers in normal or exponential representation, integers, and outsi numbers. Only decimal and integer representation will now be considered. The normal decimal representation of a number will hengelouth be referred to as "Floating Point." As integer represented without a decimal point will be raited "Floating Point." As integer represented without a decimal point will be raited "Float Point." As integer reposter without a decimal point will be raited "Float Point." As integer reposter without a decimal point will be raited "Float Point." As integer reposter without a decimal point will be raited as a specific number to called the Field Width.

The M columns available for numerical input of data can be givened into any ocate of member of licids of any width destrict. The source dask must contain a FORMAT atstement which instructs the computer in which way it should interpret the data card. The specification in the FORMAT statement for the floating-point format is

hyw En

where n is the number of identical fields to be read (the first held begins with column i !); F is the rode letter for Hoating-point representations who the field width; and n is the number of places to the right of the decimal point

The appointment of a fixed-point number in a FORMAT statement is of the form

wla

where a and ware as previously defined and I is the code-letter for livedpoint representation. In parabing a disa card with a number, the sign percodes the muribor, or flucting point number can be guiched anywhere within
the field. With a fixed-point number the number must be at the right size of
the field; otherwise, the blank spaces at the right end of the field, residing
the computer as course, would after the mimber by powers of ten. In both
representations, if the number is my as mursing, the computer assumes
that the number is positive.

Only single-line format for numerical quantities will be described. The source card for a FORMAT statement might be punched as follower:

Language

Cobunita 2 -

Column I coward to 12

sinsfil

A Statement Number located anywhere in this this teld, an integral number with-out sign.

design

might be plinched as follows:

PORMAT (3 P S.S. F S.A. IZ. MA).

This FORMAT statement could be used to specify reading of a data card. The data card would have 3 fields of 6 columns, a field of 5 columns, a field of 2 columns, and 3 fields of 4 columns. The spelling, commas, and parentheses are essential parts of the statement and in all other types of statements to be described.

If a Fortran statement requires more space than the columns from 7 to 72, a statement can be continued on a second card by punching column 6. Up to 9 continuation cards can be used. With the exception of zero, what is punched in column 6 is immaterial and can even be the same on all continuation cards in the entire program, since the computer merely reads the cards in the order in which they are stacked in the source deck. Only one statement can be punched on a single card. The statement itself need not start in column 7, and any number of blanks can be inserted anywhere in a statement. The statement number is an integral number without a sign and can be located anywhere in the columns from 2 to 5. Statement numbers uniquely identify statements and need not be in any numerical sequence.

Numerical quantities in FORTRAN can be numerals or identified with a letter name. A floating-point constant has a decimal point whereas a fixed-point constant must not. The literal expression for a number can consist of a series of not more than 6 characters, which can be both letters and numerals with certain restrictions. The first character must be a letter. A fixed-point number name must begin with letters I, J, K, L, M, or N, and floating-point number names must not begin with these letters. A good rule is to avoid the use of F as the last character to prevent possible confusion with names of functions which are discussed on page 6.

Both floating- and fixed-point variables can be subscripted with up to 3 subscripts. The subscripts must necessarily be fixed-point numbers without sign. Subscripts cannot take on zero or negative values. The following examples completely cover limitations of naming and arithmetic operations for subscripts:

algebraic	FORTRAN
aj,k,l	A(J,K,L)
xa	X(IA)
ci+2,m	C(I + 2, M)
ij,m	I(J,M)
Pzi-1	P(2*I-1)

As previously mentioned, an equation can be punched in a form which looks very much like ordinary algebra. The algebraic symbols used are +, -, *, **, /, (), and =. These symbols represent addition, subtraction, multiplication, exponentiation, division, parentheses, and equality. As an example, the algebraic expression

man recommend viscos or before or allow incommends. TAMECS and the control of the bland of the column and the column of the colu

The state of the state of the continue of the case of the case of the continue of the state of the case of the

Interest of interest and the state of the st

Both Bothing and Red point variables can be paleonized with an to a subscript of the following subscripts of the following sign. Shipping admitted take on serie or negative values in the following a quintles sompletely sover limitations of naming and arithmetics operations for subjectifies.

MARTHON NAME OF TRANS

REAL ALLE SALES

REAL CONTROL OF TRANS

LARGE SALES

LARGE S

As previously mentioned, an equation can be paided in a mera which looks very month like equinary algebra. The discount of mildle used and reflection and the parallel expenses and equalities, a vision, parestancing and equality. As an example the algebrate expenses income.

$$y = (ax^2 + bx + c)^3$$

could be written as a FORTRAN Arithmetic Statement as follows

$$Y = (A*X**2 + B*X + C)**3$$

The <u>mode</u> of all terms (i.e., fixed or floating point) on the right side of the equal sign must be the same! An exception is that an exponent of a floating-point number can and should be fixed point if it is an integral number. The mode of the left side need not be the same as the right side. When the right-side mode is different from the left-side mode, the statement implies a conversion of the right-side quantity into the mode specified by the left side. In the example above, if the left side were I, I would be equal to Y truncated, to an integer.

In writing a FORTRAN arithmetic statement, the variables, constants, and equal symbol can frequently be thought to have the same meaning they have in algebra. Actually, the variables in FORTRAN are specifications of memory locations rather than the numbers themselves. The quantity computed by using the contents of memory locations specified on the right of the equal sign are to be stored in the location specified on the left of the equal sign. Thus the equal sign really has the meaning "store." This is best illustrated with the statement

$$A = A + B$$

Algebraically this only holds true for B=0; however, in FORTRAN the results of the summing of the contents of memory locations A and B are to be stored in memory location A.

The use of parentheses in FORTRAN is very much like their use in algebra; that is, after encountering a left parenthesis, the computer will perform the indicated algebraic operations up to the right parenthesis, independently of any further algebraic operations outside of the parentheses. Thus, in the previous statement for Y, the quantity in the parentheses is computed and then raised to the third power.

In the absence of parentheses or within parentheses the order in which the algebraic operations are performed is exponentiation, multiplication or division, and finally addition or subtraction. Thus BB = A*B**C + X algebraically means the single variable called BB equals $(a \cdot b^c) + x$.

Parentheses are frequently necessary to write an arithmetic statement properly, and when in doubt it is probably wise to use them even if they are not absolutely necessary. To illustrate some cases in which parentheses are necessary, consider the following algebraic expression and their FORTRAN representation: THE RESERVE OF THE PARTY OF THE

wolled as tramers in the State and the State of the State

PARTY & RESERVE OF CHARLES

The mode of all revent [3.0., fired or floading points on the right wide at the equal sign from the tim sagnet An exception is that an exponent of a floating equal mucher can not should be investigated in the same and the cities and the right wind with the last side of the same and are in the same and the right side of the same and the right side of the same and the same and the same and the same and the same of the same o

In writing a HORTRAM articonemistion variables, variantes, variantes, constants and acquait expended to the same meaning they and acquait expended to the same meaning they have in algebra. The quantity together than the numbers appropriate the region of the quantity together appropriate to the contract and the region of the expended of the expended of the same appropriate together appropriate to the region of the expendent and the region of the contract of t

B & A & A

Argebraically this only holds true for B = 0, however, in FORTHAN the next to be and the numbried of the culture of memory locatings A and B are to be thereof is memory locating.

The use of parenth-ses if FORTAN is very mind like their see the is of algebras that is, after encountering a left parenthed in communer will perform the indicated algebras operations up to the reported parentheses. Thus, and perform any further algebras operations outside of the parentheses. Thus, is the previous statement for N. the quantity in the parentheses is computed.

In the allegate of perculpedes are within perculpedes the order in which perculped and are perfused in exponentiation, abulifying the algorithm of city and the state of the area of the a

Parantinization frequently neurosal state an arthrein state of the property and when it doubt it is probably visulouse them even if they are not absolutely mecoscopy. Tori they sto as no veres in blick parantities are newistering equitiently toward state are neurosal state.

FORTHAM representations.

$$a(b + c)$$
 $A ** (B + C)$
 $a(b \cdot c)$ $A ** (B * C)$
 $(a \cdot b)^{C}$ $(A * B) ** C$

An expression like

could have been written with extraneous parentheses as (A**B)*C. Whenever two of the characters +, -, /, *, **, appear consecutively, it is necessary to separate them with parentheses. Consider the following relationship which would not be ambiguous in algebra:

$$y = x \cdot -2. - + 4.6$$
;

this would have to be written in FORTRAN as

$$Y = X*(-2.) - (+4.6)$$

An example of an expression containing a subscripted variable might be

$$y = a_i^2 + b_j^m$$
 $Y = A(I)**2 + B(J)**M$

where the parentheses arise because of the FORTRAN subscripting notation.

FORTRAN has 5 types of functions, of which only 3 will be considered. These 3 are open (or built in) functions, closed (or library) functions, and arithmetic statement functions. These 3 functions have the same rules for their naming and for referring to them (calling).

The names of these 3 types of functions consist of 4 to 7 alphabetic or numeric characters of which the first must be alphabetic and the last is F. A function can have one or more arguments, but the function is single-valued. The mode of the arguments may be fixed or floating, and the mode of the function value can be either fixed or floating. If the value of the function is fixed point, the function name must begin with X. It is desirable to name a function with a sequence of letters which suggest its meaning.

To illustrate the naming of functions, we can consider the algebraic statement for y previously used:

$$y = (ax^2 + bx + c)^3$$

and define an Arithmetic Statement Function. The statement to be punched could be

YFUNF
$$(X) = (A*X**2 + B*X + C)**3$$

A(b + c) A ** (B + C)

A ** (B + C)

(a + b)^c (A * B) ** c

An empression like

Season . The

Could have been written with extraneous parentheses as (A**8)*C. Whenever two of the inaracters * ... / . * ... abject consecutively, if the neesssary to separate them with parentheses. Consider the following relationship
which would not be ambiguous in nighten

612 4 - S= x 5 7

this would have to be written in FORTRAN is

(a.a.+) - (.5-) - X - X

An example of an expression containing a subscripted variable

M**(L)8 + (**(L)A = Y (III + (**) + B(J)**M

where the pereminence arise because of the FORTRAN subscripting normion.

FORTRAN has 5 types of functions, of which only 5 will be considered.
These 5 are open (or built in) functions, closed (or illurary) functions, and
asilbonetic statement concions. These 5 tenedons have the same rules for
their narring and for referring to them (calling).

The names of those 3 types of directions equals of 4 to 7 shouldering of names characters of which the last in a name of characters of which the first in a single read the last in the function can have one or over at generate, but the function is single readed. The mode of the arguments may be fixed at floating, and the notice of the function value can be higher fixed or floating. If the value of the function is fixed point, the function harms much begin with a decirable to name a function with a sequence of lasters which engaged its meaning

To illustrate the naming of implions, we can consider the algebraic statement for y providedly used.

(+ + 2d + 2d) =

and deline as Arthuredy Statement Punction. The statement to be punched acted be

The name YFUNF is followed by its argument X in parentheses. YFUNF has a value which is floating point (no X at start of name). Its argument on the left side clearly shows that the argument is floating point and, we note that to the right side of the equal sign the statement is correctly written in the same mode. As defined above, YFUNF has the argument X and has A, B, and C as parameters. If the left side of the statement had been written YFUNF(X,A,B,C), the function would have 4 arguments.

To have the argument floating and the function value fixed point, the statement would be

$$XYFUNF(X) = (A*X**2 + B*X + C)**3$$

If we wished to have a fixed-point argument and floating-point function, we might write

YFUN2F (I) =
$$(JA*I**2 + JB*I + JC)**3$$

Note that the absence of X at the beginning of the name means that the value of the function is floating point. The argument in parentheses is fixed point, and to the right of the equal sign the arithmetic manipulations have the fixed-point variable I and the 3 fixed-point coefficient JA, JB, and JC to give a fixed-point-mode arithmetic expression consistent with the mode of the argument of the function in parentheses on the left side of the = sign.

The fourth possibility is to have fixed-point argument and fixed-point value for the function:

$$XYFUN2F(I) = (JA*I**2 + JB*I + JC*I)**3$$

In FORTRAN as in ordinary algebra we can put in any letters or numbers for the argument of a function, and so if the 4 foregoing functions are defined in a program, they may later be called by arithmetic statements as follows:

ALPHA = 2.5 + YFUNF (7.2)

BETA = 6.3 + YFUNF (Y(K))

GAMMA = 5.75 + YFUN2F(J)

DELTA = 20.5 + YFUN2F(31)

IOTA1 = 20 + XYFUNF(2.6)

IOTA2 = 25 + XYFUNF (Z)

LAMDA1 = 31 + XYFUN2F(IT)

LAMDA2 = 436 + XYFUN2F(39)

The name YEURE is followed by the Expument of in namentary in argument, and the control of the c

I de his est to a registrate floating and the fraction value fix d point, like

AND DESCRIPTION OF THE PROPERTY OF THE PROPERT

nality or runner and all best team are rander based at budate aw H will be at the budate aw H will be such as the budate are this in so

PART (A) A STATE (CAU) = (D) TANDEN

Note that the absence of M at the beginning of the name miners that the of the forest that the plant of the forest of the forest

The fourth possibility is to have fixed point or exament and fund about

XVERDINERU) = (LATELLE EBET = TONICE

In FORTICAL as an artificate and as a public of the second letters or the billion for the artificial for the following the continues of the following for the following following for the following following for the following following

ALIGNA - 2.5 - ARBUA

(A)YI-WAUSYAN AND - ATRE

A LIZE

TANDAY . SEE XYFENZEGT

TANDALE & ALE + XYTUNE TOP

The <u>built-in</u> and <u>library</u> functions are defined within the 704 and, just as the foregoing functions were utilized in arithmetic statements, so library functions can be used by appropriately indicating the function name and its arguments. Seven library functions with floating-point values and arguments are

LOGF	natural log of the magnitude of the argument
SINF	sine (angle in radians)
COSF	cosine (angle in radians)
EXPF	exponential function
SQRTF	Plus square root of the magnitude of the argument
ATANF	$-\pi/2$ < arctangent in radians < $\pi/2$
TANHF	hyperbolic tangent

Table I, taken from the FORTRAN manual, lists the built-in functions in the 704 FORTRAN, and their use is probably self-explanatory.

Table I

		Number		Mode of		
Type of Function	Definition	of Args	Name	Argument	Function	
Absolute value	Arg	1	ABSF XABSF	Floating Fixed	Floating Fixed	
Truncation	Sign of Arg times largest integer ≤ Arg	1	INTF XINTF	Floating Floating	Floating Fixed	
Remaindering (see note below)	Arg ₁ (mod Arg ₂)	2	MODF XMODF	Floating Fixed	Floating Fixed	
Choosing largest value	Max (Arg ₁ , Arg ₂ ,)	≥2	MAXOF MAXIF XMAXOF XMAXIF	Fixed Floating Fixed Floating	Floating Floating Fixed Fixed	
Choosing smallest value	Min (Arg ₁ , Arg ₂ ,)	≥ 2	MINOF MINIF XMINOF XMINIF	Fixed Floating Fixed Floating	Floating Floating Fixed Fixed	
Float	Floating a fixed number	1	FLOATF	Fixed	Floating	
Fix	Same as XINTF	1	XFIXF	Floating	Fixed	
Transfer of sign	Sign of Argz times Arg1	2	SIGNF XSIGNF	Floating Fixed	Floating Fixed	
Positive difference	Arg ₁ , -Min (Arg ₁ , Arg ₂)	2	DIMF XDIMF	Floating Fixed	Floating Fixed	

NOTE: The function MODF (Arg₁, Arg₂) is defined as Arg₁ - [Arg₁/Arg₂] Arg₂, where [x] = integral part of x.

Reproduced by permission from IBM Reference Manual - 704 FORTRAN Programming System, @ 1958, 1959, 1961 by International Business Machines Corp.

The built-in and library functions are defined within the The Ball, fuel at the locations of the Ball of the Ball

grammars and to shalldam sit to ter eraspe and the Arthur

and the land was well.

Table 1, reten from the FOR IRAN magnin help the ball in functions with the

at A de Boen-han land and an Stable per ma set in the

Reginaling a program and the material Manual - 104 FIR 18.850 Programming Systems (D. 1926, 1926, 1930, 1940, by international United States and Manual Manu

In summary with regard to the use of the 3 classes of functions described, the part of a statement which calls the function consists of the function name followed by a list of the arguments separated by commas and in parentheses, as for instance,

SOMEF (X, Y, Z(I))

Whenever a variable is subscripted, it is necessary to assign a sufficient number of memory locations for the array. The assignment of memory locations is done with a DIMENSION statement.

For instance, if in a problem we have the subscripted variables A(I), B(I,J), and C(I,J,K), which would be 1-, 2-, and 3-dimensional arrays, respectively, where I takes on values from 1 to 5, J values from 1 to 4, and K values from 1 to 3, then 5 memory locations are required for A(I), 20 locations for B(I,J), and 60 locations for C(I,J,K). The DIMENSION statement for this specific example would be

DIMENSION A(5), B(5,4), C(5,4,3)

In the source deck, the DIMENSION statement must appear before the first statement which refers to the subscripted variable.

A simple problem will now illustrate the use of the statements which have been described and to introduce a few other statements. The problem will be to have the program read 5 data cards, each of which has a single value of an angle in radians, and to have the computer evaluate the sine, cosine, and tangent of these angles, and form a four-column tabulation consisting of the angle, its sine, cosine, and tangent.

Columns 2 to 5	Columns 7 to 72
correction in 1	DIMENSION A(5)
2	FORMAT (F10.5)
Man workers the 3 man of	FORMAT (4F10.5)
4	READ 2, $(A(I), I = 1,5,1)$
by the section 5	DO 9 I = 1,5,1
6	SI = SINF (A(I))
7	CO = COSF (A(I))
8	TA = SI/CO
9	PRINT 3, A(I), SI, CO, TA
10	STOP
11	END (0,1,0,0,1)

In summary with regard to the use or then classes of uncrious dessorthed, the part of a manment which walls the function construe of the braiding cause followed by a list of the arguments separated by comman and in parent

SOMEE (X, Y, Z(I))

Whenever a variable is subscripted, it is necessary to assign a suffreed number of memory locations for the arts. The assignment of memory locations is dead with a DIMENSION statement.

For nature, this a problem we have the subscripted variables.A[1], one G(1,1,1), which would be 1s , 2s , and 3 dimensional arrays respectively, where I takes on values from I to 5, 5 values from 1 to 4, and M values moth I to 3, then 5 memory locations are returned for A[1], 30 formulations for G(1,2,2). The High High Selection statement

DIMENSION AISH B(8,4), C(8,4,5)

in the source deck, the DIMENSION scalement mist appear before the filtest statement which refers to the subscripted vertable.

A simple problem will now plujerate the use of the statements which have been accordant. The problem have been accordant which are statement. The problem will be to have the group and to have the carde existent of an angle in radians, and to have the computer evaluate the sine content and tempera of their angle, and town of the cardes the sine, costing at the radians and range it.

Columns 2 to 5: Columns 7 to 7

(E) A MOTSWEIMIG

ROBOWAL CRIDER

FOR NAT 44 ET 0 43

BEEVE S (VIII) 1 - 1 .2.11

THE PARTY OF

100-1200-001

OO US - AT

ATTOOLS THE ATTORNEY

HONE !

altality) area

The statements have all been numbered for ease of description, though only statements 2, 3, and 9 require statement numbers for execution of this program. Statement I says that a "subscripted variable A, i.e., Angle, will require 5 memory locations." Statement 2 says; "a single floating-point number occupying a field of width 10 with 5 places to the right of the decimal point." This statement will be used by statement 4 to instruct the computer how to read data cards. Statement 3 says; "4 floatingpoint numbers, each occupying a field of width 10 with 5 places to the right of the decimal point." This statement is used by statement 9 to instruct the computer how to type the value of angle, sine, cosine, and tangent. Statement 4 has not been previously explained; it can be read "Read data cards according to the specification of statement 2 (i.e., FORMAT statement); what is being read defines the value of subscripted variable A(I) where the index I begins with 1, and runs to 5 in steps of 1." This read statement, at object time, will cause 5 data cards to be read because of the way the index I and FORMAT were specified. The numbers read are stored in the computer memory.

Statement 5 has not previously been explained. It can be read "do the instructions specified by the statements following up to and including statement 9, and then repeat this series of instructions as governed by the index I which is to take on the values 1 to 5 in unit steps." Thus in our case the DO statement specifies that statements 6 to 9 are to be executed 5 times; each time it repeats the index I is increased by unity. We can say the DO statement has defined a loop which is to be circled 5 times. The last number (1) which defines how I is incremented need not appear in statements 5 and 4 because its absence implies that it is unity. Within the DO loop the index number I is available for use in fixed-point arithmetic expressions. After the DO loop is satisfied, the index number I is no longer defined.

Statements 6 and 7 are simple arithmetic expressions which define floating-point variables in terms of the library functions SINF and COSF. Note that in naming the variables SI and CO, it is forbidden to name them SIN or COS, since a variable must never have a name which is that of a defined function without the terminal F.

Statement 8 is an Arithmetic Statement that TA the tangent is the sine divided by the cosine.

Statement 9 has not been previously explained; it can be read "Print on the output printer (typewriter) according to the specifications of FORMAT statement 3, the 4 quantities A(I), SI, CO, and TA."

If it is desirable to print out the 5A(I) values before entering the DOloop, a PRINT Statement could be placed between statements 4 and 5:

64

The sixt ments have all been immbered for each of deep interest of the statement of the property distributed and 9 require statement should be a required of the property distributed being a superty localities. Statement rooms for the action of the property distributed by statement rooms for the statement of the

Statement 5 has not previously been explained in such be said the the trainment of the analysis of the statements following up in any he quing statement of the statement of the statement of the greather is an intervention as greather by the unit statement is to be statement be such as the intervention of the said of the statement has a specific the statement between the statement has defined a bong which is to be civaled being the statement has defined a bong which is to be civaled being the statement has defined as the statement of the statement has defined how it is increased to be civaled being the statement of the statement is a validated to the statement of the statement o

Statements & and I are simple arithmetic expressions which felles (nather spines which lead Const.)

Doubles spines virtables in terms of the ubrace (nathous SIRE has Const.)

Very that in naturing the variables SI and Co. It is forblocked in many them.

SIN or COS, since a variable must never have a many which is that of differential formula. If,

Statement S lawn Arrigmetic Statement that TA the tangent to the ame distribut by the position

Statement 9 has not been previoually excellent of it can be read Perest on the corpus period (representation) according to the amendmentation of representation of the comment is the 4 quantities A(I), all CU, and TA.

n er e destrable to proce out the Ea(1) valued limites and the Colorest limites and the Colorest

This would print the 5 values of A(I) on 5 successive lines. Note that the PRINT statement above and the READ statement 4 have the nature of a DO loop. Note that the list of variables in PRINT and READ statements requires parentheses only if the statements define an index and require looping.

Statement 10 will be executed after the DO loop is satisfied. Its effect at object time is to stop the computer.

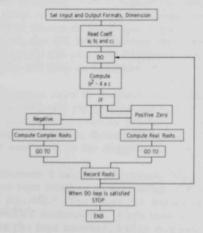
The <u>END Statement</u> is the physically last card in the source deck. It essentially separates independent programs whose source decks are put into the card reader at the same time during compiling. At the Argonne National Laboratory, the numbers 01001 are the normal END statement specifications. Note that the computer performs the program in the sequence of the source deck cards except for the change caused by statements such as DO.

FORTRAN has a number of different conditional branching statements, of which only one will be considered. The statement

can be read "If the variable or expression A is negative, skip in the program to statement 5; if it is zero, skip to statement 20; and if it is positive, skip to statement 30." Care must be exercised in testing floating-point expressions for zero because of truncation and round-off errors which are normally introduced into a calculation. Fixed-point arithmetic, on the other hand, is exact, and the skip on zero presents no difficulties.

The unconditional branch statement is GO TO.

The following problem will serve to illustrate the use of IF and GO TO statements. The problem is to factor 5 quadratic equations by means of the quadratic formula. The roots of the equations can be either real or complex. The block diagram of the program is as follows.



This would pelot the 5 values of All on 7 species lines. The first the PAINT statement shows as a DO PAINT statement shows and the READ statement & bireship as the FRINT and READ statements requires parents and require the statements delines a sold and a require the statements delines a sold at a require the statements delines as a sold at a square the statements delines as a sold at a square the statements delines as a sold at a square the statement of the statement

Statement 10 will be executed after the DO loop is satisfied; the

The END Statement is the present which send in the sauge deriv.

It passestially supplied as independent programs whose source decide are put less the correct reader of the same time decided and the nervest END statement some interestings. It is somewhat is supplied the program in the computer performs the program in the computer performs the program in the source deck carries accept for the change caused by statements as DO

TORTEN best Laureley of Militarest conditional Branching statements of which cate and bearing at the constant of The statement

TF (A) 5,20,30

can be read "If the variable or expression A is regarive, also in Tox present to statement for it is zero; step in expression for the bestilve, skip in tratement for "Gare must be ascenised in leating floating point expressions for early because of trategration and remarkable in a constitution into a calculation. Tixed-point erituments, on the filler hind, is exact, and the alte on sero presents to difficulties.

The unconditional branch vistament to GO TO

The following problem will serve to illustrate the use of IT and GO
TO statements. The problem is to factor 5 questions by means of
the quadratic formula. The roots of the equations can be sides real or com-



```
DIMENSION A(5), B(5), C(5)
       FORMAT (3 F 10.5)
    FORMAT (4 F 10.5)
       READ 5, (A(I), B(I), C(I), I = 1,5)
           DO 9 I = 1.5
                 D = B(I)**2 - 4.*A(I)*C(I)
                    IF(D)10,4,4
10
               R1 = -B(I)/(2.*A(I))
               R2 = R1
               AII = SQRTF (D) /(2.*A(I))
              AI2 = -AI1
                      GO TO 9
             R1 = (-B(I) + SQRTF(D))/(2.*A(I))
               R2 = (-B(I) - SQRTF(D))/(2.*A(I))
              AI1 = 0.
               AI2 = 0.
              WRITE OUTPUT TAPE 2, 6, R1, AI1, R2, AI2
                    END FILE 2
                    STOP
                    END (0,1,0,0,1)
```

In the foregoing program, the IF statement tests if the discriminant b^2 - 4ac is negative, zero, or positive. If it is negative, the program beginning with statement 10 is followed, since the roots are complex. After the complex roots are found, the GO TO statement tells the computer to execute statement #9, which records the roots on magnetic tapes, and then the program continues until the DO loop is satisfied.

If the discriminant is zero or positive, the roots are real, and the IF statement says that the program for finding the real roots begins with statement #4. If the discriminant was zero or positive, statement 9 follows immediately after the roots have been determined, and no GO TO statement is necessary to have the next step in the program be the recording of the roots.

The output statement 9 has not previously been explained. It can be read: "Write on output tape number 2, according to FORMAT statement 6, the numbers R1, AI1, R2, AI2." The rules for listing the output quantities are the same as for PRINT statements. Each tape unit has a ten-position switch which identifies the tape number, but at Argonne, tape number 1 is

TOKENSION A(s), B(S), C(S)

FORMAT (1 7 10 5)

ECHAAT (4 7 10 5)

LEAD S. (A(t), B(t), C(t), I; 1.5

DO S I = 1, S

DO S I = 1, S

 $D = B(t)^{**2} + 4 * A(t)^{*}C(t)$ D = B(t)/(2, *A(t))

All = SORTE (D)/(2.*A(I))

IJA= = SIA

La = (-H(I) = SORTE(D))/(2.*A(I))

to = LLA.

WRITE OUTPUT TAPE 2. 6 RJ. AH. RZ. AK

EMB FILE

tron in his man

In the foregoing program, the IF statement tests if the discrimnant be - fac is negative, zero, or positive. If it is negative, the program beginning with statement 10 is followed, since the roots are complex. After the complex roots are found, the GO TO statement tells the computer to execute statement #3, which recercis the roots on magnetic tapes, and then the program continues until the DO loop is satisfied.

If the discriminant is zero or positive, the roots are real, and the IF statement says that the program for fluding the real roots begins with statement of if the discriminant was zero or positive, statement 9 follows the mediately after the roots have been determined, and no CO TO statement is necessary to have the next star in the orogram be the recording of the motor.

The untput statement 4 has not proviously been explained. It can be read: "Write on output tape number 2, according to FORMAT statement 6, the numbers R1. All R2 All." The rides for listing the output quantities are the same as for PRINT statements. Each tape unit has a ten-position switch which identifies the tape number 1 to at Argonna, tape number 1 to

not to be used. Since magnetic tape recording is much faster than the "On line" printing (on the Printer directly associated with the 704), the time charges on the 704 are reduced. The recorded output tape is later read and printed by an "Off line" printer for which there is no additional charge at the Argonne computer center.

The <u>END FILE</u> statement marks the output tape number 2 after all the output has been recorded. The END FILE mark on the tape is used to stop the off-line printer. At the Argonne computer, the convention is not to include a REWIND statement for the output tape, so that the computer operator can record the output from several problems on the same reel of tape.

Thus far, a major part of the description has been concerned with introducing various types of statements and describing the form in which they are written; some further mention must be made of restrictions.

Of the statements described, the FORMAT and DIMENSION statements are called non-executable statements, that is, they do not cause an operation to be performed. The first statement after a DO statement must not be a non-executable statement. The last statement of a DO loop must not be a statement which transfers control, as does GO TO. When a control transfer statement is the last executable statement in a DO loop, the non-executable dummy statement CONTINUE is made the last statement of the DO loop.

The successive statements between the DO statement and the last statement of the DO loop define the range of the loop. Frequently, other DO statements occur within a DO loop. When this occurs, the range of the earlier DO statement encountered must extend up to or beyond the range of any DO loops defined within the range of the first DO loop. This says that DO statements must be <u>nested</u>; there cannot be just a partial overlap of ranges of DO loops.

Transfer into a DO loop from statements outside of the range of a DO loop can only be made to the DO statement itself, and not to any other statement in the range of the DO loop. With a single DO loop, transfers are permitted to statements within the range of the loop. If there are nested DO statements in a program, any transfers to the inner DO statements can only be to the DO statements itself so as not to violate the rule in regard to transferring stated above. The calling of any function or subroutine involves a transfer out and then back into a program; this sort of transfer is permitted within the range of a DO loop provided the subroutine returns to the same part of the loop and that the subroutine does not alter the value of indices. It should also be emphasized that no statement within a DO loop must redefine the indices. It should be noted that all restrictions apply to entering DO loops and not to exiting.

not to be ined. Since inspectic tage recording as much faster than the "Online" printing (on the Fringer directly associated with the 704), the tage changes on the 704 are reduced. The recorded output tage is later read and printed by an "Off line" printer for which there is no additional charge-atthe Arrowse computer center.

The END WILE statement marks the output tape number 2 after all the count has been necessary. The END FILE mark on the tape is need to stop the off-line printer. At the Argone computer, the convention is not to include a REWIND statement for the output tape, so that the computer opportunity can record the output from saveral problems on the same real of these.

Thus fur, a major part of the description has been concerned with unroducing various types or statements and describing the form in which they are wrotten; some further mention until be made of restrictions.

Of the statement's described, the FORMAT and DIMEYSION states and motera are called non-executable statement after a 100 statement must operation to be performed. The first last statement after a DO statement must not be a figure executable statement. The first statement of a DO loop must not be a statement which industry control, as does DO TO When a confrol threefer statement is the last are made in a DO loop, the non-executable dummy statement CONTINUE is made the last statement of the DO loop.

The successive streetents between the DC statement sad the last streetent of the DC statement of the DC loop define the senge of the last streetent of the DC statement occur within a DC loop. When the occurs the tange of the Color of the vond the range of the DC loop. This says of may DC loops defined within the carge of the DC loop. This says that DC loops defined within the carge of the DC loop. This says that DC loops of the DC loop.

Francher into a made to the DO statements outside oblige range of a set of most can only be made to the DO statement least, and not be any ellent of the DO loop. With a ringle DO loop, transfers are permitted to electricals within the range of the loop. If there are needed not be transfer as a program, any transfers to the loop. If there are needed not be transfer in the transfer in the range and the box of the loop and the face of the range of any function of subsending the transfer of any fine the range of a DO loop provided the subsending of the loop and that the enhancement when the raise of indices.

It should also be supplied that no statement with: a DO loop must redefine the indices. It should be noted that all restrictions apply to agreeting the indices. It should be noted that all restrictions apply to agreeting

Some mention should be made of size limitations of numbers. A fixed-point number must be less than or equal to 2¹⁷. A floating-point number can be between 10⁺³⁸ and 10⁻³⁸ or zero; however, the computer carries only 8 significant digits!! A greater number of significant digits on a data card are not read.

The maximum length of a FORTRAN statement is 660 characters or blanks, and requires 9 continue cards. The maximum number of characters and blanks which can be printed on a single line is 120. The length of a single tape record on output tape is also 120 characters and blanks. A tape record is printed off-line as a single line.

Whenever an Arithmetic Statement Function is defined, it must precede the first executable statement in the program.

It is frequently convenient to include comment card in the source deck. Comment cards are punched with a C in column #1, and any comments such as program name, date, and program author are punched in columns 2 through 80 on any number of cards in succession. Comment cards at the beginning are a convenient way for the operator to identify the program being compiled, since the comment cards and all the statement cards are printed during compiling. Comment cards produce no effect upon the object deck.

In writing FORTRAN, care must be taken not to confuse the letter O with the numeral zero, which are punched differently on a card. The letter O should be written as ϕ . The letter I and the number I can also cause trouble although this is due more to careless writing. The printing on a punched card is different for I and I, so no confusion results.

Acknowledgments

The author appreciates the suggestions and comments of numerous people in the Applied Mathematics and International Institute divisions. The author is particularly grateful to Mr. Burton Garbow of the Applied Mathematics division for his detailed review of the original manuscript. Suggested rephrasing of certain parts of the text which might be more satisfying to an experienced 704 user have not been incorporated in the final form of the paper in the hope that its form is more comprehensible to a novice.

Some thindian should be reade for the limitations of minners - since -

The maximum liested of a FORTRAN statement as 500 claracters or blanks, and requires, a seather cards. The maximum number of characters and blanks which the back had in 1200. The length of a single tage record on support of the cards of th

Collegence on Artificial Conference and Artificial frames pre-

List frequently convented to include comment ears to the control of the control o

In artiting FORTRAN, care must be taken not to conjust this letter O with the admired account of the letter O Columbia be written as Q. This letter I had the multiper F consider a consider the confider of the province of a consider which is different for F state, so no confider in results.

Addinowladgerers

The Author appreciates the suggestions and convenient of mimerous property in the Applied Mathematics and international finantiate distributed for the Applied Mathematics and authority parties at the Applied Mathematics distributed for the Applied Mathematics distributed for the Applied Mathematics of the Contract of the Internation of the Contract of the Contract

APPENDIX

Some additional information on procedures of the Argonne computer center will now be given. This will be helpful in carrying through a program to the final solution of the problem.

The program is most conveniently written on the form shown in Figure 1. Data are written on the form shown in Figure 2. The forms are available in the stationery section of the Building 203 stockroom, Room R129 on the main floor.

After preparing the written program, the source cards can be punched in Room C055 in the basement of Building 203. After an initial keypunch practice period, source cards should be punched by the girls in Room C055. Arrangements for card punching should be made with Lead Data Transcription Operator, Room C055, Extension 2876. Other card punches are located in various other divisions of the Laboratory, including one in Building 25A, Extension 2294. Generally, at each card-punching machine a supply of blank IBM cards is available. Figure 3 shows a sample source card and a data card.

The IBM 704 computer is currently installed in Room D001 in the basement of Building 203. In the same room is a 1401 computer which is programmed to check cards of the source deck. The checking of a source deck is referred to as "Pre-processing." An applied mathematics memo is available which includes a list of errors checked by the preprocessor and the code name by which they are identified. There is no additional time charge for preprocessing and, since it greatly increases the probability that the program will work on the 704, all source decks should be preprocessed. The 1401 is scheduled for preprocessing from 1030 to 1115 and from 1500 to 1545; if requested, the 1401 operator will check the results and, if it appears that the program can be compiled, will place the deck in the next compiling batch. Compiling is accomplished from 1200 to 1300 (if requested) during the day; also, compiling is done at night for pickup the next morning.

An application for authorization to obtain use of the 704 must be filed with the Applied Mathematics Division in the division office, Room B153, Building 203, by means of form AMD 6. AMD then establishes a job number and job time cards are sent to the computer room. Argonne National Laboratory employees should become acquainted with the policy of their Division in relation to authorization of computer use. Use of the 704 by persons not employed by Argonne is authorized by the Division sponsoring the person.

The specific time at which the computer is to be used is scheduled at the computer room or by calling the scheduler at Extension 2877. Requests for time of 15 min or less from 0830 to 1200 are made after 1200 of

Some interioral internation on procedures of the Arrenas comparer content will now his given. This will be belowd in carrying the program to the high column of the problem.

The program is most conveniently writteness the form save in Figure 1. Date are windles of the form shown in Figure 2. The forms are available in the stationery section of the Dutleting 20 s sockream. Room RIFO on the ment floor.

Alter Dispering the written program the score error on he purched to the control of the control

The IEM 704 computed in carried to an ablication of the computer of the control o

An application for authorization to obtain when of the fifty must be that with the Applied Manhamatics Invision in the division office Room Middle of the first of the AMID then Variable a substance of the Manhamatics of the Computer room. Argoric Myllogist Edge of the Language of the Myllogist Room and John Middle of the Myllogist Room of the Middle of the Myllogist Room of the Middle of the Myllogist Room of the Myllogist Room.

The specific time at which the computer is to be used as scheduled at the computer according the scheduler at Extension 2877. Requests for time at 15 min or issuefrom 0850 to 1200 acc made size 1200 of

the preceding day; requests for time from 1300 to 1600 are made prior to 1200 of that day. The schedule is written on the blackboard in the computer room. Time on the 704 can be scheduled from 1700-2400 for periods of time greater than 15 min.

Whenever time is scheduled, a time card and a set of instructions for the computer operator must be provided. Form AMD 5 is used for the instructions. The necessary information on the form is the following:

- (A) The nature of the run, that is, whether it is compiling, computation, or both.
- (B) Does the program have on-line printing or punching?
- (C) Does it have output on magnetic tape and, if so, which tape switch number setting should be made? It should be specified if a single or double copy should be produced by the off-line printer. The single and double copy are referred to as 1 or 2 part paper, respectively. If the tape is to be saved, what identification should it have on its label? (Label information should include the name of the job, the name of the person who requested that the tape be saved, and the date after which the tape can be reused.) If no labeling information is provided, it is assumed that the tape need not be saved.
- (D) Does the program use any data cards?
- (E) What are the sense switch settings? (This introduction has not covered sense switches, so one merely says they are all up.)
- (F) What is estimated duration of the run? The operator should be informed when to stop a run, particularly if the program is new, to avoid wasting time on a program which may malfunction. For programs which have been previously run successfully, the operator can be told to allow the program to run to completion and the time required can be predicted with reasonable accuracy.

After compiling, the source and object decks and the printed sheets produced by the compiler are on distribution shelves in D-001 where they can be picked up.

The off-line printing from magnetic tapes is routinely done during the day within an hour and a half of the completion of the computer run. The printed output can be picked up from the shelves adjacent to the 1401 computer. the proceeding day; requests for time right 100 to 1000 are made prior to 1200 on the blackdown in the computer room. There are the roll was be sensed about 1700-4700 for periods of time greater than 15 mag.

Whenever time is athended, a time card and a set of instructions for the computer operator must be provided. Form AMD 5 or used for the instructions. The correspondy tofo matter on the form is the following:

(A) The milities of the rost, that is, whether it is complising, compared assembly on the complision, work oit.

(B) Dock the program bare on line printing or punching

10) Deck it have comes on magnetic maps and, if so, which tage

I will be manager setting should be maded it at tide be specified.

It will be manager of double copy should no produced by the off-sine.

I will paper. The angle and double copy are referred to as i on

I will paper. respectively it the tage is to be seved, what

I the bill setting the angle of the copy at the mane of the person which the

sequence that the tage he seved and the date after which the

At these can be recently it in a take ing and one date after which the

Debt the program set mangare and assu. (C

(b) What are the sente switch settings? (This introduction has

(F) What is estimate of direction of the court the operator about the information of the court of the program of the information of the court of the program of the court of the program of the base than provide the program of the base than provide the program of the court of of the

After compiling the abarde and object decks and the bribled shells produced by the compiler are negligibled shelly as in D-00 i where they can be preked up.

The day within an hour and a half pithe completion of the computer was.

The printed output can be ut took by from the shalves adjacent to the 1401 computer.

	PROBLEM	DATE	PAGE	OF	PROGRAMMER
+ FOR COMMENT					
STATEMENT NUMBER		FORTRAN STATEMENT			IDENTIFICATION
5 6	7			72	73 8
IM S					
1					
1					
			- 124 AN		
Partie -	-				
+	-				
1					
	The second				
!	T		11 15		-
	16.1		7 21 129	_	
1					
	THE THE PARTY OF THE PARTY.			TARREST TO	

Figure 1

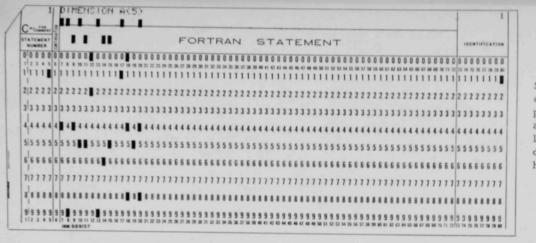
IBM Fortran Programming Form (reproduced by permission of International Business Machines Corporation.)

704 INPUT DATA

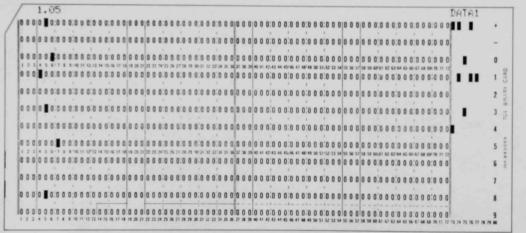
COST CODE

ROGRAM	PROBLE	EM	ORIGINATO	OR	DATE	PAG	E OF
	1	2	3	4	5	6 7	
23456789	0 1 2 3 4 5 6 7 8	90123456789	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0123454789	0 1 2 3 4 5 6 7 8 9 0	123454789
							1 111111
			111111111111111111111111111111111111111				1 1111111
						111111111111111111111111111111111111111	
							1 1111111
TTT	ullunu	""		ши.		The state of the s	
					LELLLELLE		
23456789	0 1 2 3 4 5 6 7 8	90123456789	0 12 3 4 5 6 7 8 9	0123454700		0 1 2 3 4 5 6 7 8 9 0	1 111111
1652 h 2 Ly	1	2	3	4	5	0 1 2 3 4 5 6 7 8 9 0 1	23 4 5 6 7 8 9

Figure 2



Sample FORTRAN Statement card. Note difference in printing of capital letter I and the numeral one. The letter O and zero can only be distinguished by the punched holes.



Sample data card. Note that a different type of card is normally used for the source cards and data cards.

Figure 3

