AN INTERPRETATIVE FLOATING DECIMAL SYSTEM FOR THE IBM TYPE 650

G. R. Trimble, Jr.

E. C. Kubie

Introduction

This floating decimal system will perform 18 basic operations using a floating decimal number system by means of interpretative programming. It was designed with coding convenience as a prime objective. This is illustrated by the fact that one of the basic operations is essentially a vector by vector multiplication.

Floating decimal instructions, that is, instructions which are to be interpreted, have a negative sign. A floating decimal instruction consists of a two digit operation code, a 4 digit address specifying the location of the first factor, a 4 digit address specifying the location of the second factor (if required) and another 4 digit address specifying the location of the third factor (if required). It is a variable address system in that only as many addresses as are needed for the particular operation are required. Thus, some operations use only one address, some require two addresses and others require 3 addresses.

Floating decimal instructions will be taken from consecutive memory locations. If an instruction requires one or two addresses, the instruction is stored in one memory location. If three addresses are required, the third address is stored in the memory location immediately following the one containing the instruction.

Should a positive instruction appear, it is interpreted as a normal 650 instruction and subsequent instructions are not interpreted. Thus, upon occurrence of a positive instruction the 650 operates in its usual D-I mode. This continues until an instruction address of 0026 is given which causes control to return to the interpretative routine. The program will return to the floating decimal mode of operation at the point of departure.

For example, consider the following sequence:

Location	Contents
n	floating point instruction (-)
n+1	floating point instruction (-)
n+2	floating point instruction (-)
n+3	normal 650 instruction (+)

(The contents of n+3 are interpreted as a normal 650 instruction including the instruction address for sequencing. Normal execution of instructions continues until 0026 is used as an instruction address, at which time the program returns to the floating decimal mode of operation beginning with the instruction in location n+4).

Number Form

The number form is as follows:

```
 ┌──────────────────────── Exponent +50
 │        ┌─────────────── Mantissa
 xx   xxxxxxxx±
```

Instruction Form

The instruction form is as follows:

```
              ┌──────────────────────── floating point command (01-18)
              │   ┌──────────────────── 1st address
              │   │     ┌────────────── 2nd address
location i    xx xxxx xxxx-
location i+1  oo xxxx oooo-
```

i+1 has this form only if the third address is needed, otherwise, the form of i+1 is the same as the form of i.

Floating Decimal Accumulator

A floating point accumulator referred to as K is used for accumulation and accumulative multiplication. It does not need to be addressed in operations which make use of it. For example, operation 01 ($A+B \longrightarrow K$) uses two addresses, A and B. The sum is automatically stored in K as the result of this operation.

Operations

Below is a list of the operations with their codes, addresses required and estimated average time of execution.

Code and Addresses	Operation	Estimated Average Time
01, A, B	$A+B \longrightarrow K$	62.8
02, A	$A+K \longrightarrow K$	54.7
03, A, B	$A-B \longrightarrow K$	63.1
04, A	$K-A \longrightarrow K$	59.5
05, A, B, C	$A+B \longrightarrow C$	81.8

Code and Addresses	Operation	Estimated Average Time
06, A, B, C	$A-B \longrightarrow C$	96.2
07, A, B, C	$AxB \longrightarrow C$	84.8
08, A, B	$AxB \longrightarrow K$	78.4
09, A, B, C	$A/B \longrightarrow C$	92.2
10, A, B	$A/B \longrightarrow K$	72.2
11, A, B	BR MIN A	28.4
12, A	BR	18.7
13, A, B	BRNZ A	30.8
14, A, B	$(AxB)+K \rightarrow K$	102.5
15, A, B	$K-(AxB) \rightarrow K$	102.5
16, A, B	$\sqrt{A} \longrightarrow B$	157.9
17, A, B, C	$\sqrt{A^2+B^2+C^2} \longrightarrow K$	416.3
18, A_1, B_1, n	$\sum_{i=1}^{n} A_i B_i \rightarrow K$	(32.4+92.3n)

Explanation of Programs

1. General Interpretation

The general interpretation routine takes instructions from consecutive memory locations and analyzes them to see if they are normal 650 instructions or if they must be interpreted. If the instruction is a normal 650 instruction, it is executed as such. If the instruction is to be interpreted, the routine obtains the factor at address A and stores it in location 0037. Control is then transferred to the proper sub-routine. The constant in location 0193 facilitates use of the translating routine TR1. The amount of translation is placed in the instruction address positions of this constant. This amount is then added to the operation code and control is transferred to the translated sub-routine.

2. Addition Sub-Routine

The addition sub-routine adds the factors in locations 0037, in which A is stored, and 0057, which is the floating decimal accumulator K. The result is normally stored in the floating decimal accumulator K.

3. Multiplication Sub-Routine

The multiplication sub-routine multiplies the factors in 0037, which is A, and 0089, in which B is placed. The product is normally stored in 0037.

4. Division Sub-Routine

The division sub-routine divides the factor in 0037 which is A, by the factor B, which had previously been placed in the lower accumulator. The result is normally stored in 0057 which is the floating decimal accumulator.

The following sub-routines interpret the second and third addresses if required, obtain the necessary factors and modify the addition, multiplication and division sub-routines so as to store the result in the desired location. In some cases, they also modify portions of other sub-interpretative routines.

5. Interpretation of 01

This sub-interpretative routine obtains the factor B and transfers control to the addition sub-routine.

6. Interpretation of 02

Since the factors are already in place all that is necessary for this sub-interpretative routine is to transfer control to the addition sub-routine.

7. Interpretation of 03

This sub-interpretative routine obtains the factor B, reverses its sign and transfers control to the addition sub-routine.

8. Interpretation of 04

The sign of the factor A is reversed and control is transferred to the addition sub-routine.

9. Interpretation of 05

Since the addition sub-routine makes use of the floating decimal accumulator K, the contents of K must be temporarily transferred to another location, and returned after the operation has been completed. This sub-interpretative routine obtains B, modifies the addition sub-routine to store the result in location C and transfers control to the addition sub-routine. After completion of the operation, control is returned to the sub-interpretative routine and the addition sub-routine is restored to normal.

10. Interpretation of 06

This sub-interpretative routine modifies the sub-interpretative routine, "Interpretation of 05" so that the sign of factor B is reversed.

11. Interpretation of 07

This sub-interpretative routine obtains the factor B, modifies the multiply sub-routine to store the product in location C and transfers control to the multiply sub-routine. After completion of the multiplication, control is returned to this sub-interpretative routine and the multiply sub-routine is restored to normal.

12 Interpretation of 08

The factor B is obtained, the last instruction of the multiply sub-routine is modified to store the product in K and control is transferred to the multiply sub-routine. After completion of the multiplication, control is returned to the sub-interpretative routine "Interpretation of 07" which restores the last instruction of the multiply sub-routine to normal.

13. Interpretation of 09

The factor B is obtained, the last instruction of the divide sub-routine is modified to store the quotient in C and control is transferred to the divide sub-routine. After completion of the division, control is returned to this sub-interpretative routine and the last instruction of the divide sub-routine is restored to normal.

14. Interpretation of 10

The factor B is obtained and control is transferred to the divide sub-routine.

15. Interpretation of 11

The sign of the factor in location A is examined. If the sign is minus, control is transferred to location B by modifying the general interpretation routine. Sequencing of instructions then continues in the normal fashion beginning with B. If the factor in location A is positive or its mantissa is zero, sequencing of instructions continues in the normal fashion.

16. Interpretation of 12

The general interpretation routine is modified so that the next instruction is taken from location A. Sequencing of instructions then continues in the normal fashion beginning with A.

17. Interpretation of 13

The factor in A is examined and if its mantissa is not zero, the general interpretation routine is modified to take the next instruction from location B. Sequencing of instructions then continues in the normal fashion beginning with B. If the factor in A is zero, sequencing of instructions continues in the normal manner.

18. Interpretation of 14

The factor B is obtained, the last instruction in the multiply sub-routine is modified to store the product in 0037, which is A, and control is transferred to the multiply sub-routine. After completion of the multiplication, control is returned to the sub-interpretative routine, the last instruction of the multiply sub-routine is restored to normal and control is transferred to the addition sub-routine which computes the desired sum.

19. Interpretation of 15

The sign of the factor B is reversed and control is transferred to the sub-interpretative routine "Interpretation of 14".

20. Interpretation of 16

This sub-interpretative routine computes the required square root and stores it in location B. The initial approximation is $X_0 = (1+4A)/(4+A)$, obtained from the RAND Corporation "Approximations in Numerical Analysis" form 15S, Notes.

21. Interpretation of 17

Factor B is obtained and temporarily stored in 0265. Factor A is squared and stored in 0057 by the multiplication sub-routine. Factor B is then squared and added to the square of A. Factor C is then obtained, squared and added to $A^2 + B^2$. Finally, control is transferred to the square root sub-routine and $\sqrt{A^2+B^2+C^2}$ is computed and stored in 0057 which is K.

22. Interpretation of 18

Operation 18 is a vector by vector multiplication. Address A_1 and address B_1 are the addresses of the first elements of each of the vectors. Succeeding elements of the vectors are then taken from consecutive memory locations starting with the initial locations A_1 and B_1. The third address, n, indicates the number of elements in each vector.

Use is made of the sub-interpretative routine "Interpretation of 14" to perform accumulative multiplication. As each pair of factors are multiplied together and added to the previous sum, n is reduced by 1 and zero tested. When n has been reduced to zero, the last two factors have been multiplied and the operation is complete.

Storage

This system requires 466 7 storage locations. It uses locations 0000 through 0465. Every location within this block is used, thus making it easy to incorporate this program with other programs. The translating routine, TR1 may be used to translate this program to any desired block of memory locations. It should be translated by an even amount however to preserve the even-odd conditions.

[illegible] 0047, 0173, [illegible], 0353, 0372 [illegible] 0423 [illegible]

[illegible] 0017 — 0024 [illegible]

[illegible]

IBM
Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: General Interpretation WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0026	RAU	60	0029	0092	
0092	AU	10	0099	0118	Increase i to i+1
0118	ST U	21	0029	8001	
8001	RAL	65	i+1	0439	Pick up instruction i+1
0439	BRMIN	46	0025	8002	Is instruction i+1 to be interpreted?
0025	LD	69	0028	0031	
0031	ST DA	22	0035	8001	Put data at address A in location
8001	LD	69	A	0455	0037.
0455	SLT	35	0002	0334	
0334	ST D	24	0037	0040	
0040	AU	10	0193	8003	Transfer to proper sub-routine.
8003	NO OP	00	0000	00fc	
0029	RAL	65	i	0439	
0099		00	0001	0000	Constants
0028	LD	69	0000	0455	
0193		00	0000	0000	

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: (A) + (K) ⟶ K Subroutine   WRITTEN BY: ____________

(A) = $a_1 A_1$, (K) = $a_2 A_2$, a=xx, A= x. xxxxxxx±

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0054	RAL	65	0057	0061	0057: K
0061	SLT	35	0002	0067	
0067	STU	21	0073	0027	$a_1 \to 0073$
0027	STL	20	0081	0034	$A_1 \to 0081$
0034	RAL	65	0037	0041	0037: A
0041	SLT	35	0002	0048	
0048	STU	21	0052	0055	$a_2 \to 0052$
0055	STL	20	0059	0062	$A_2 \to 0059$
0062	RAABL	67	8003	0069	
0069	SABL	18	0073	0077	$\lvert a_2\rvert - \lvert a_1\rvert$. Modify SRD instruction
0077	SLT	35	0004	0087	for case where $0 < \lvert a_2 - a_1\rvert < 10$
0087	LD	69	0090	0043	
0043	ST DA	22	0098	0051	
0051	BRNZ	45	0104	0056	Is $a_2 = a_1$?
0056	RAL	65	0059	0078	$a_2 = a_1$. Thus $A_3^1 = A_2 + A_1$
0078	AL	15	0081	0038	
0038	BRNZU	44	0091	0044	Is $\lvert A_3^1\rvert \geq 10$?
0091	SRD	31	0003	0101	$\lvert A_3^1\rvert \geq 10$. Thus $A_3 = 0.1\, A_3^1$
0101	SLT	35	0002	0058	
0058	BR MIN	46	0111	0112	Is $A_3 < 0$?
0111	SL	16	0065	0070	$A_3 < 0$. Increase exponent by 1
0070	SABL	18	0073	0127	
0127	SLT	35	0008	0045	Put exponent in normal positions
0045	AL	15	8003	0053	(2 high order digit positions)
0112	AL	15	0065	0120	$A_3 > 0$. Increase exponent by 1
0120	AABL	17	0073	0127	
0044	BRNZ	45	0049	0053	$\lvert A_3^1\rvert < 10$. Is $A_3^1 = 0$?
0049	RAU	60	8002	0066	
0066	SCT	36	0000	0088	$0 < \lvert A_3^1\rvert < 10$. Thus A_3 equals A_3^1
0088	SU	11	8002	0105	shifted and counted.
0105	RAL	65	8003	0116	
0116	BR MIN	46	0070	0120	Is $A_3 < 0$?
0104	LD	69	0057	0060	
0060	SRT	30	0005	0074	$a_2 \neq a_1$. Is $\lvert a_2 - a_1\rvert \geq 10$?
0074	BRNZ	45	0080	0079	
0080	BR MIN	46	0033	0084	$\lvert a_2 - a_1\rvert \geq 10$. Is $a_2 > a_1$?
0033	RAL	65	8001	0053	$a_1 > a_2 + 10$. Thus $a_3 A_3 = a_1 A_1$
0084	RAL	65	0037	0053	$a_2 > a_1 + 10$. Thus $a_3 A_3 = a_2 A_2$
0079	BR MIN	46	0032	0039	$0 < \lvert a_2 - a_1\rvert < 10$. Is $a_2 > a_1$?
0032	RAL	65	0059	0064	
0064	SRT	30	0002	0072	$a_1 > a_2$. Thus $A_3^1 = A_1$ + shifted A_2
0072	LD	69	0081	0098	
0098	SRD	31	(0000)	0071	
0071	SLT	35	0002	0030	Compute A_3^1
0030	AL	15	8001	0038	
0039	LD	69	0052	0068	
0068	ST D	24	0073	0076	
0076	RAL	65	0081	0036	$a_2 > a_1$. Thus $A_3^1 = A_2$ + shifted A_1
0036	SRT	30	0002	0046	Also, use a_2 to compute a_3
0046	LD	69	0059	0098	

PROBLEM: (A) + (K) ⟶ K Subroutine   WRITTEN BY:

(A) = a_1 A_1, (K) = a_2 A_2 a = xx, A = x.xxxxxxx±

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0053	ST L	20	0057	0026	Store sum in K
0090	: 31 0000 0071				Constants
0065	: 00 0000 0001				

PROBLEM: (A) x (B) ⟶ A Subroutine   WRITTEN BY:

(A) = m_1 M_1, (B) = m_2 M_2, m=xx, M=x.xxxxxxx ±

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0134	RAL	65	0037	0141	0037: A. Separate exponent and
0141	SLT	35	0002	0097	mantissa
0097	STL	20	0151	0106	M_1 ⟶ 0151 M_1 =x.xxxxxxxxoo
0106	RAU	60	8003	0063	
0063	SRT	30	0002	0119	m_1 - 50
0119	RAABL	67	8002	0128	
0128	SL	16	0131	0085	
0085	AU	10	0089	0093	0089: B. Test to see if M_2<0
0093	BR MIN	46	0096	0147	
0096	SU	11	8001	0103	
0103	SL	16	8001	0161	M_2 < 0
0161	SLT	35	0002	0117	Put M_2 in upper accumulator and
0117	AU	10	8002	0075	M_3' = m_1 + m_2-50 in lower accumulator.
0075	RSU	61	8003	0138	Acc: ooMMMMMMMM/mmoooooooo
0138	SRT	30	0002	0095	
0147	SU	11	8001	0155	
0155	AL	15	8001	0113	M_2 ≥ 0.
0113	SLT	35	0002	0121	Put M_2 in upper accumulator and
0121	AU	10	8002	0115	m_3^1 = m_1 + m_2 -50 in lower accumulator
0115	RAU	60	8003	0138	
0095	MPY	19	0151	0082	M_3' = M_1 x M_2
0082	BR MIN	46	0135	0086	
0135	SL	16	0139	0094	
0086	AL	15	0139	0094	Round M_3^1 in 7th decimal place
0094	LD	69	8003	0100	
0100	SLT	35	0002	0107	
0107	SRT	30	0001	0163	Test to see if M_3^1 is of the form
0163	SCT	36	0001	0170	ox.xxxxxxxx or xx.xxxxxxxx
0170	BROV	47	0123	0126	
0123	SRT	30	0009	0181	
0181	SRD	31	0002	0102	
0102	ST DA	22	0205	0108	
0108	ST IA	23	0211	0114	M_3^1 is of the form ox.xxxxxxxx
0114	RAL	65	8001	0171	Round M_3^1 in 8th decimal place
0171	BR MIN	46	0124	0125	to get M_3.
0124	AL	15	0178	0083	
0125	SL	16	0178	0083	
0083	ST L	20	0037	0026	
0126	RAL	65	8001	0042	M_3^1 is of the form xx.xxxxxx. Thus,
0042	BR MIN	46	0000	0050	m_3^1 = m_3 +1
0000	SL	16	0109	0083	
0050	AL	15	0109	0083	
0131	: 5000000000				Constants
0139	: 5000000000				
0178	: 0000000005				
0109	: 0100000000				

* In floating decimal mult. 0181 will result in overflow in case of all 9's. This will invalidate the 2 subsequent Store Address instructions.

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: (A)/(B) → K Subroutine WRITTEN BY:

(A) = $d_1 D_1$, (B) = $d_2 D_2$ d=xx, D= x. xxxxxxx

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0213	SLT	35	0002	0169	
0169	STU	21	0174	0177	$d_2 \rightarrow 0174$
0177	STL	20	0081	0184	$D_2 \rightarrow 0081$
0184	RAL	65	0037	0191	
0191	SLT	35	0002	0148	
0148	STU	21	0059	0263	$d_1 \rightarrow 0059$
0263	RAU	60	8002	0221	
0221	SRT	30	0001	0227	$D_3^1 = D_1/D_2$
0227	DIV RU	64	0081	0464	
0464	BR MIN	46	0159	0209	
0159	SL	16	0162	0167	Round D_3^1 in 8th decimal place.
0209	AL	15	0162	0167	
0167	SLT	35	0001	0223	Test to see if rounded D_3^1 is of the
0223	BRNZU	44	0130	0228	form x. xxxxxxxxx or . xxxxxxxxxx
0228	BR MIN	46	0132	0182	
0132	AL	15	0185	0189	D_3^1 is of the form o. xxxxxxxxx, thus
0182	SL	16	0185	0189	round D_3^1 in the 9th decimal place
0189	SRT	30	0002	0145	to get D_3.
0145	SLT	35	0002	0152	
0152	BR MIN	46	0255	0156	
0255	SABL	18	0059	0164	
0164	SL	16	0217	0321	
0321	AABL	17	0174	0179	$d_3 = d_1 - d_2 + 50-1$
0156	AABL	17	0059	0214	
0214	AL	15	0217	0371	
0371	SABL	18	0174	0179	
0130	SRT	30	0003	0140	D_3^1 is of the form x. xxxxxxxxx, thus
0140	SLT	35	0002	0149	$D_3 = D_3^1$
0149	BR MIN	46	0153	0154	
0153	SABL	18	0059	0313	
0313	SL	16	0166	0321	$d_3 = d_1 - d_2 + 50$
0154	AABL	17	0059	0363	
0363	AL	15	0166	0371	
0179	RAU	60	8002	0137	
0137	SRT	30	0002	0143	Put d_3 in normal position.
0143	AU	10	8002	0202	
0202	STU	21	0057	0026	
0162	:0000000050				
0185	:0000000450				Constants
0217	:0000000049				
0166	:0000000050				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 01: (A) + (B) → K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0001	SLT	35	0002	0157	
0157	LD	69	0110	0165	
0165	ST DA	22	0073	8001	Put B in 0057
8001	RAL	65	B	0453	
0453	STL	20	0057	0054	
0110	: 6500000453				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 02: (A) + (K) → K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0002	NO OP	00	0000	0054	

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 03: (A) - (B) → K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0003	SLT	35	0002	0160	
0160	LD	69	0215	0168	
0168	ST DA	22	0073	8001	Put -B in 0057
8001	RSL	66	B	0453	
0453	STL	20	0057	0054	
0215	: 6600000453				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 04: (K) - (A) → K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0004	RSL	66	0037	0133	Put -A in 0037
0133	STL	20	0037	0054	

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 05.
(A) + (B) → C

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	CODE	ADDRESS DATA	INSTRUCTION	REMARKS
0005	SLT	35	0002	0212	
0212	LD	69	0216	0220	Modify instruction to get B
0220	ST DA	22	0273	0176	
0176	RAU	60	0029	0183	
0183	AU	10	0186	0192	Increase i+1 to i+2
0192	AL	15	8003	0199	
0199	LD	69	0252	0206	
0206	ST DA	22	0059	8001	Get third address (C) from i+2
8001	AABL	17	i+2	0463	
0463	LD	69	0057	0262	Put K in 0265 temporarily
0262	ST D	24	0265	0218	
0218	SL	16	8003	0175	Store i+2 in 0029
0175	ST D	24	0029	0187	
0187	LD	69	0142	0196	Modify add routine to store sum
0196	ST DA	22	0053	0273	in C.
0273	RAL	65	B	0453	Get (B) and transfer to add
0453	ST L	20	0057	0054	routine
0188	LD	69	0265	0195	Return K to 0057
0195	ST D	24	0057	0194	
0194	LD	69	0197	0150	Restore last instruction of Add
0150	ST D	24	0053	0026	routine to normal.
0216	: 6500000453				
0186	: 0000010000				
0252	: 1700000463				Constants
0142	: 2000000188				
0197	: 2000570026				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 06:
(A) - (B) → C

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	CODE	ADDRESS DATA	INSTRUCTION	REMARKS
0006	SLT	35	0002	0264	Modify interpretation of 05 to
0264	LD	69	0267	0220	obtain -B
0267	: 6600000453				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 07: (A) x (B) → C

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0007	SLT	35	0002	0266	
0266	RSL	66	8002	0225	
0225	AU	10	0029	0233	Modify instruction to get B
0233	LD	69	0236	0190	
0190	ST DA	22	0243	0246	
0246	RAL	65	8003	0203	
0203	AL	15	0207	0261	
0261	LD	69	0314	0268	Increase i+1 to i+2. Get C and
0268	ST DA	22	0073	8001	
8001	SU	11	i+2	0456	modify last instruction of multiply
0456	ST L	20	0029	0239	
0239	AU	10	0242	0198	routine to store product in C
0198	ST U	21	0083	0243	
0243	RAL	65	B	0456	Get B and store it in 0089. Go
0456	ST L	20	0089	0134	to multiply sub-routine.
0454	LD	69	0226	0230	Restore last instruction of multiply
0230	ST D	24	0083	0026	routine to normal
0236:	6500000465				
0207:	0000010000				
0314:	1100000456				Constants
0242:	2000000454				
0226:	2000370026				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0 PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 08: (A) x (B) → K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0008	SLT	35	0002	0315	
0315	LD	69	0219	0172	
0172	ST DA	22	0277	8001	Get B and store it in 0089
8001	LD	69	B	0462	
0462	ST D	24	0089	0244	
0244	LD	69	0158	0129	Modify last inst. of mult. routine
0129	ST D	24	0083	0134	to store product in K, then go to
					0454 to restore last inst. of mult.
					routine to normal.
0219:	6900000462				
0158:	2000570454				Constants

IBM
Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 09:
(A)/(B) ⟶ C

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0009	SLT	35	0002	0317	
0317	RSL	66	8002	0276	
0276	AU	10	0029	0234	Modify instruction to get B
0234	LD	69	0237	0240	
0240	ST DA	22	0243	0248	
0248	AU	10	0201	0208	
0208	RAL	65	8003	0316	Increase i+1 to i+2, get C and
0316	AU	10	0269	0224	modify last instruction of divide
0224	ST L	20	0029	0232	routine to store quotient in C.
0232	LD	69	0285	0238	
0238	ST DA	22	0241	8001	
8001	SU	11	i+2	0461	
0461	ST U	21	0202	0243	
0243	RAL	65	B	0213	Get B and go to divide routine.
0293	LD	69	0296	0249	Restore last instruction of divide
0249	ST D	24	0202	0026	routine to normal.
0237:	6500000213				
0201:	0000010000				
0269:	2100000293				Constants
0285:	1100000461				
0296:	2100570026				

IBM
Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 10:
(A)/(B) ⟶ K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0010	SLT	35	0002	0222	
0222	LD	69	0237	0290	Put B in lower accumulator and go
0290	ST DA	22	0243	8001	to divide routine
8001	RAL	65	B	0213	
0237	6500000213				Constant

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 11: Branch Minus   WRITTEN BY: ____________

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0011	SRT	30	0002	0417	
~~0417~~	~~AU~~	~~10~~	~~0037~~	~~0378~~	Test for minus. If plus, do not branch
0378	BR MIN	46	0382	0026	
0382	SLT	35	0002	0339	
0339	SRT	30	0002	0397	Test for zero. If zero, do not branch
0397	BRNZU	44	0352	0026	
0352	SLT	35	0004	0420	
0420	LD	69	0373	0426	Modify 0029 and get next instruction
0426	ST DA	22	0029	8001	
8001	RAL	65	B	0439	
0373:	6500000439				Constant

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 12: Branch   WRITTEN BY: ____________

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0012	SRT	30	0002	0420	
0420	LD	69	0373	0426	Put A in 0029
0426	ST DA	22	0029	8001	
0373:	6500000439				Constant

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 13: Branch Non-Zero   WRITTEN BY: ____________

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0013	SRT	30	0002	0423	
~~0423~~	~~AU~~	~~10~~	~~0037~~	~~0431~~	
0431	BR MIN	46	0382	0336	Put −|A| in upper accumulator
0336	SU	11	8001	0443	
0443	SU	11	8001	0382	
0382	SLT	35	0002	0339	
0339	SRT	30	0002	0397	Test for zero. If zero, do not branch
0397	BRNZU	44	0352	0026	
0352	SLT	35	0004	0420	
0420	LD	69	0373	0426	Modify 0029 and get next instruction
0426	ST DA	22	0029	8001	
8001	RAL	65	B	0439	
0373:	6500000439				Constant

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 14: (A) x (B) +(K) ⟶ K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0014	SLT	35	0002	0422	
0422	LD	69	0375	0278	
0278	ST DA	22	0081	8001	Put B in 0089
8001	RAL	65	B	0358	
0358	ST L	20	0089	0392	
0392	LD	69	0398	0357	Modify last instruction of multiply
0357	ST D	24	0083	0134	routine to store product in 0037
0399	LD	69	0402	0355	Restore last instruction of multiply
0355	ST D	24	0083	0054	routine to normal
0375:	650000	0358			
0398:	200037	0399			Constants
0402:	200037	0026			

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 15: (K) - (A) x (B) ⟶ K

WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0015	SLT	35	0002	0421	Modify "Interpretation of 14" to
0421	LD	69	0424	0278	obtain -B
0424:	660000	0358			Constant

IBM
Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 16: $\sqrt{(A)} \rightarrow B$ WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0016	LD	69	0326	0329	
0329	SLT	35	0002	0335	Modify instruction to store square
0335	ST DA	22	0241	0306	root in B
0306	RAL	65	0037	0323	
0323	SLT	35	0002	0279	Separate a_1 A_1. Store A_1 in 0283
0279	ST L	20	0283	0286	
0286	RAL	65	8003	0343	
0343	AU	10	8001	0251	
0251	AU	10	8001	0210	$1/2\ a_1$
0210	AU	10	8003	0367	
0367	AU	10	8002	0325	
0325	SRT	30	0001	0281	
0281	SU	11	8003	0289	Is a_1 even or odd?
0289	SLT	35	0001	0245	
0245	BRNZU	44	0250	0200	
0250	RAU	60	8001	0307	
0307	AU	10	0260	0415	a_1 even. $a_2 = [1/2\ a_1] + 25$
0415	SRT	30	0002	0322	
0322	ST L	20	0277	0330	$a_2 \rightarrow 0277$
0330	RAU	60	0283	0287	
0287	SRT	30	0002	0294	
0200	RAU	60	8001	0257	
0257	AU	10	0260	0365	a_1 odd. $a_2 = [1/2\ a_1] + 25$
0365	SRT	30	0002	0272	
0272	ST L	20	0277	0280	$a_2 \rightarrow 0277$
0280	RAU	60	0283	0337	
0337	SRT	30	0001	0294	
0294	AL	15	8003	0301	
0301	AL	15	0204	0259	Store A in 0265.
0259	ST U	21	0265	0318	Store 4+A in 0273.
0318	ST L	20	0273	0376	
0376	RAU	60	8003	0284	
0284	AU	10	8001	0291	Compute 1+4A
0291	AU	10	8003	0299	
0299	AU	10	0253	0258	
0258	SRT	30	0001	0270	$X_0 = (1+4A)/(4+A)$
0270	DIV RU	64	0273	0440	
0440	AU	10	0265	0369	
0369	SL	16	8002	0327	Test to see if A is zero.
0327	BRNZ	45	0380	0274	
0380	DIV RU	64	8001	0460	
0460	AL	15	8001	0413	
0413	SRT	30	0001	0320	
0320	AU	10	8002	0379	$X_1 = 1/2\ (X_0 + A/X_0)$
0379	AL	15	8001	0387	
0387	AL	15	8002	0295	
0295	AL	15	8003	0303	
0303	RAL	65	8002	0311	
0311	AU	10	0265	0370	
0370	SL	16	8002	0381	

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 16: $\sqrt{(A)} \rightarrow B$ WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0381	DIV RU	64	8001	0451	
0451	AL	15	8001	0304	
0304	SRT	30	0001	0312	
0312	AU	10	8002	0374	$X_2 = 1/2\ (X_1+A/X_1)$
0374	AL	15	8001	0282	
0282	AL	15	8002	0292	
0292	AL	15	8003	0300	
0300	RAL	65	8002	0310	
0310	AU	10	0265	0419	
0419	SL	16	8002	0377	
0377	DIV RU	64	8001	0452	
0452	AL	15	8001	0256	
0256	SRT	30	0001	0364	$X_3=1/2\ (X_2+A/X_2)$
0364	AU	10	8002	0324	
0324	AL	15	8001	0331	
0331	AL	15	8002	0340	
0340	AL	15	8003	0450	
0450	RAL	65	8002	0409	
0409	AU	10	0265	0180	
0180	SL	16	8002	0144	
0144	DIV RU	64	8001	0458	
0458	AL	15	8001	0459	$X_4 = 1/2\ (X_3+A/X_3)$
0459	SRT	30	0001	0235	
0235	AU	10	8002	0146	
0146	AL	15	8001	0405	
0405	AL	15	8002	0231	
0231	AL	15	8003	0247	
0247	RAU	60	8002	0305	
0305	SCT	36	0000	0361	Normalize the square root and
0361	BROV	47	0366	0366	reset overflow circuit.
0366	SRT	30	0002	0274	
0274	AU	10	0277	0241	Put exponent in normal position
0241	ST U	21	B	0026	and store result in B
0236:	2100000026				
0260:	-0000000025				Constants
0204:	0400000000				
0253:	0100000000				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 17: $\sqrt{(A)^2 + (B)^2 + (C)^2} \longrightarrow K$

WRITTEN BY: ____________

LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
	ABBRV.	CODE	DATA	INSTRUCTION	
0017	SLT	35	0002	0275	
0275	LD	69	0037	0136	
0136	ST D	24	0089	0342	
0342	LD	69	0345	0298	
0298	ST DA	22	0351	8001	Prepare to compute $(A)^2$ and
8001	RAL	65	B	0408	
0408	LD	69	0427	0430	put $(A)^2$ in 0057
0430	ST D	24	0083	0346	
0346	ST L	20	0265	0134	temporarily put (B) in 0265.
0360	RAL	65	0265	0333	
0333	ST L	20	0089	0297	Prepare to compute $(B)^2$ and
0297	LD	69	0254	0308	
0308	ST D	24	0083	0141	put $(B)^2$ in 0037
0341	LD	69	0395	0349	Prepare to compute $(A)^2 + (B)^2$ and
0349	ST D	24	0053	0054	store it in 0057.
0362	RAL	65	0029	0383	
0383	AL	15	0186	0391	
0391	LD	69	0394	0347	
0347	ST DA	22	0351	0354	Increase i+1 to i+2. Get C
0354	AU	10	8001	0411	
0411	ST L	20	0029	8003	and prepare to compute $(C)^2$
8003	RAL	65	i+2	0466	
0466	SL	16	0338	8002	and put it in 0037
8002	RAL	65	C	0437	
0437	ST L	20	0089	0396	
0396	LD	69	0302	0368	
0368	ST D	24	0083	0141	
0390	LD	69	0393	0348	
0348	ST D	24	0053	0309	Prepare to compute $(A)^2+(B)^2+(C)^2$
0309	LD	69	0226	0429	and store it in 0037
0429	ST D	24	0083	0054	
0328	LD	69	0332	0288	Restore last instruction of add
0288	ST D	24	0241	0344	routine to normal
0344	LD	69	0197	0350	Prepare to compute $\sqrt{(A)^2+(B)^2+(C)^2}$
0350	ST D	24	0053	0306	
0345:	6500000408				
0427:	2000570360				
0254:	2000370341				
0395:	2000570362				
0394:	6500000466				
0338:	6500000437				Constants
0302:	2000370390				
0393:	2000370328				
0332:	2100570026				

IBM Trade-Mark

IBM TYPE 650 PROGRAM SHEET

FORM NO. 22-6181-0
PRINTED IN U.S.A.

PROBLEM: INTERPRETATION OF 18: $\sum_{i=1}^{n} (A_i) \times (B_i) \rightarrow K$ WRITTEN BY:

LOCATION OF INSTRUCTION	OPERATION ABBRV.	OPERATION CODE	ADDRESS DATA	ADDRESS INSTRUCTION	REMARKS
0018	SRT	30	0002	0425	
0425	SU	11	0029	0384	Store (18 $A_1 B_1$) in 0389
0384	ST L	20	0389	0442	
0442	RSL	66	8003	0401	
0401	ST U	21	0057	0412	Put zeros in K
0412	AL	15	0416	0432	
0432	LD	69	0385	0388	Increase i+1 to i+2 and get n
0388	ST DA	22	0241	8001	
8001	SU	11	i+2	0457	
0457	ST L	20	0029	0433	
0433	SU	11	0386	0441	Store n-1 in 0386
0441	LD	69	0444	0447	Modify last instruction of add
0447	ST D	24	0053	0356	routine to return control to 410
0356	ST U	21	0414	0434	
0434	RAL	65	0389	0445	Prepare to get B_1
0445	SLT	35	0004	0422	
0410	RAU	60	0414	0418	Test to see if n has been reduced
0418	BRNZ	45	0122	0438	to zero.
0122	SU	11	0428	0436	
0436	AABL	17	0389	0446	Decrease n by 1. Increase A_i and
0446	AL	15	0449	0359	B_i by 1.
0359	ST U	21	0414	0435	
0435	ST L	20	0389	0448	
0448	LD	69	0403	0406	
0406	ST DA	22	0059	8001	Get A_{i+1} and prepare to get B_{i+1}
8001	LD	69	A_{i+1}	0407	
0407	ST D	24	0037	0404	
0404	SLT	35	0004	0422	
0438	LD	69	0197	0400	Restore last instruction of add
0400	ST D	24	0053	0026	routine to normal.
0416:	0000010000				
0385:	1100000457				
0386:	0000010000				
0444:	2000570410				Constants
0428:	0000010000				
0449:	0000010001				
0403:	6900000407				