F E R R A N T I   L T D.

FERRANTI ATLAS COMPUTER

Programming Exercises

LIST CS 349
SEPTEMBER 1963

CONTENTS

ATLAS PROGRAMMING EXERCISES

1. INTRODUCTION

These exercises are intended to be used in conjunction with the ATLAS provisional Programming Manual CS 348. The relevant chapter of the manual is indicated at the beginning of each set of examples.

Solutions to the examples have been provided where these were considered necessary. In many cases there is more than one possible solution, the solutions provided should therefore be regarded only as guides.

2. NOTES ON SOLUTIONS

Any working space used in the store must be addressed.

The entry point of each solution should be indicated by an arrow and an address.

The contents of any address are denoted by that address enclosed in round brackets.

Thus 11.4 refers to address 11.4
(11.4) refers to the contents of address 11.4

All examples are independent unless specifically stated otherwise.

3. ELEMENTARY EXAMPLES

3.1 Accumulator operations (see chap. 3).

'x' represents a floating point number held in 509.0
Similarly $y = (510)$
$z = (511)$

1) Place x in store line 500.0, y in 501.0, and z in 502.0.

2) Place -x in store line 500.0, and clear (i.e. set to zero) store lines 509.0 and 510.0.

3) Replace x by y, y by -z, and z by x.

4) Calculate 2x - y and leave the result in Am.

5) Put 17x - 12y into line 512.

6) Replace x by xy, y by yz, and z by zx.

7) Write $x^2 + y^2 + z^2$ to line 500 and $x^2 - y^2 + z^2$ to line 501.

8) Compute $(x - y)^5$ and write the result to line 512.

9) Leave in the accumulator the quantity $-(x - y)(y - z)(z - x)$.

10) Replace x by $\frac{y}{z}$, y by $\frac{z}{x}$, and z by $\frac{x}{y}$.

11) Compute $\frac{x - y}{y - z}$ and place the quotient in line 40.

12) Evaluate $\frac{-(x - y)(y - z)(z - x)}{(x^2 + y^2 + z^2)}$ and leave the result in the accumulator

13) Write $\sqrt{\frac{x}{y}}$ in place of z.

14) Write $\frac{(x^2 + y^2 + z^2 + 2xy + 2yz + 2xz)}{2x(x + y + z)^2}$ into store line 2.

3.2 B-register operations (see chap. 4).

a represents the contents of half word 510.4
Similarly $b = (511.0)$
and $c = (511.4)$

1. Place a in B1, b in B2 and c in B3.

2. Place a in store line 512.0, b in 512.4, and c in 513.0.

3) Place 2b in B1, 3b in B2, and 7b in B3.

4) Replace a by a + b + c.

5) Replace b by a - b + c.

6) Replace a by -b, b by -c, and c by -a.

7) Calculate 4(a + b) - 3c and leave the result in B20.

8) Write 11.4 to B1, (11.4) to B2, and -25 to B3.

3.3 <u>Common extracode operations</u> (see chap. 7).

Suppose x = (10), y = (11), z = (12) are floating point numbers. Store any constants required.

Place in store line 13 each of the following.

1) lx - yl

2) Z tan X

3) arctan (1 + integral part of x^2)

4) $(x^2 + y^2)^{-z}$

4. <u>MODIFICATION COUNTING AND TESTING</u>

4.1 <u>Set I</u> (see chap. 4).

1) Place 0, 1, 2, 3, ..., 99 respectively in the half words with addresses A4+i, i = 0 to 99.

2) Compute in the accumulator the product of the n fullword numbers at successive addresses from A10, where preset parameter P26 has been set to the value n($\geq$2).

3)a. The full words A17+i, i = 0 to 13+P17, contain half word numbers xi, yi, where xi is the left half word. Place the sum xi - yi in half word A49, assuming P17$\geq$-13.

b. Place in B6 the address of the first full word such that xi = yi, assuming P17$\geq$-13.

4) Add to the accumulator the full word numbers stored at addresses 60, 66, 72, 78, ..., 114.

5) Clear the half words with addresses 1D1, 3D1, 6D1, 10D1, 15D1, ..., 4950D1.

6) Replace b5 by $b5^2$ and b6 by $(b6)^{2^{P7}}$, P7 being a positive integer-valued parameter. Place 1 + [b1/b2] in half word A6, assuming $b_2 \neq 0$.

7) Suppose that half word A6+i contains ai, i = 0 to n. Compute $a_0x+a_1x^2+a_2x^3 \ldots +a_nx^{n+1}$, where x is contained in B12 and n = P41(≥0).

8) The consecutive half words beginning at A77 hold several numbers less than P13; count them in B30, stopping when the address of such a number becomes greater than A90 (≥A77).

9) Among the consecutive full words beginning at A13 there is one whose left half word contains its own address. Place the address of this full word in B90.

10) Place the sum of ai/bi in A10. where ai, bi are 100 half word numbers stored consecutively from A1 and A2 respectively. Finally, place the integer part of that sum in B3.

4.2 Set II (see chap. 4).

Any constants required such as a zero in the clearing operations of Group A, should be stored and labelled.

Group A

1) Clear the 100 full words beginning at address A3.

2) Clear the k full words beginning at address S_0 where preset parameter P7 = k and S_0 is stored in B12.

3) Clear the k_i (i = 1 to 10) full words beginning at S_i, where the k_i are positive integers stored in consecutive half words from A1, and the S_i are full word addresses stored in consecutive half words from A2.

4) Beginning at half word A7 clear those half words which contain +1.0 until 1000 half words are clear.

Group B

1) Consider the following scheme, defining a 'chain list' containing the half word numbers x_o, x_i, .., x_n

Non consecutive full word addresses	Contents of left half word	Contents of right half word
a_0	x_0	a_1
a_1	x_1	a_2
a_2	x_2	a_3
.	.	.
.	.	.
.	.	.
a_{n-1}	x_{n-1}	a_n
a_n	x_n	a_0

Place the sum of the x's in the half word A3, given that B7 contains the address a_0.

2) Compress the x_i, given above, into the n+1 half words beginning at the half word labelled A12. Assume that the initial address a_0 = A13.

Group C

Suppose that a_i (i = 0 to n) are floating point numbers stored in consecutive full words from the full word labelled A1. Thus a_i = (iA1), i = 0 to n. Similarly, the floating point numbers b_i = (iA2), i = 0 to n. The positive integer n = P7 (preset parameter).

1) Let $a_0 b_n + a_1 b_{k-1} + \dots a_k b_0 = c_k$
Place in full word A40 the sum $c_0 + c_1 + \dots + c_n$.

2) Place in the full word labelled A3 the continued fraction.

$$a_0 + \cfrac{b_0}{a_1 + \cfrac{b_1}{a_2 + \cfrac{\ddots}{a_{n-1} + \cfrac{b_{n-1}}{a_n + b_n}}}}$$

5. SHIFTS AND LOGICAL OPERATIONS ON THE CONTENTS OF B-REGISTERS

5.1 Set I (see chap. 4)

1)a. Represent the following decimal numbers as octal numbers in B-lines.

23
-23

b. Interpret the following decimal numbers in the notation a:b.c

513
2000

2) B12 contains J737. Write down the contents of B12 in octal after each of the following series of operations have been performed.

a) 1340 12 0 3
   1343 12 0 3

b) 1340 12 0 3
   1344 12 0 3

3)a. Replace b7 by its logical binary complement

b. Place a 1 bit in B7 wherever B9 and B10 disagree.

4) Replace bits 10 - 13 of B6 with bits 13 - 16 of B9.

5. Add B30, B31 bit by bit modulo 2, multiply bit by bit modulo 2 putting the results in B1 and B2 respectively.

6. Set $b7 = \begin{matrix} 0 \text{ if } b7 > 0 \\ 1 \text{ if } b7 < 0 \end{matrix}$

7. Cause the sign bit of B71 a) to be 0
   b) to be 1

8. B3 contains a character in bits 0 - 5. Bits 6 - 23 are zero. Push this character into bits 12 - 17 of B4 pushing 1 octal digit off either end of B4.

9. Make bit 5 of B80 a (1 if b13 is odd
   (0 if b13 is even.

10. Replace every other 1 bit of B60 with a 0 starting with bit 0.

11. B12 contains the octal digits a b c d e f g h
    B13 contains the octal digits A B C D E F G H

    a) Replace the contents of B12 by A b c d e f g H

    b) Replace the contents of B12 by E F G H a b c d

    c) Replace the contents of B12 by A b C d E f G h

    preserving b13 in each case.

5.2 Set II (see chap. 4)

Group A Use only basic codes.

1) Reverse the order of the bits in B62.

2) Count in B31 the number of one bits in B17.

3) Count in B41 the number of half words (regarded as 24 bit non-negative integers) in blocks 40 to 50 which contain an exact power of 2.

4) Suppose that lists A and B of 1000 half word numbers each are stored consecutively from addresses A29 and A77. The lists are identical except where A contains a zero and B does not or where B contains a zero and A does not. Replace A with the list of all non-zero entries of the two lists; except where both entries are zero.

Group B   Extracodes 1340 - 1344 may be used.

The following exercises involve three 24 × 24 matrices whose elements are zeros and ones.
Matrix F is held in B-lines 1 to 24, where the ith row of F is the contents of B(i + 1), i = 0 to 23.
Matrix G is similarly contained in B25 to B48.
Matrix H is held in the left half words of the 24 consecutive full words with addresses iA40, i = 0 to 23.
Thus half word iA40 is the ith row of H.

1) Set F = 0 (the zero matrix).

2) Set H = I (the unit matrix).

3) Compute in B80 the trace of 6, taken modulo 2 (the trace of a matrix is the sum of its diagonal elements).

4) Copy G^t (the transpose of G) to H.

5) Compute the matrix product F × G and store it in the original location of G. The elements are to be computed modulo 2.

6. ACCUMULATOR OPERATIONS

6.1 Set I (see chap, 5)

List 1

1. 0	6. -250×2^{-14}
2. -100	7. 1/10 (specify to 13 octal digits)
3. 125×10^{-3}	8. 3×8^5
4. -2^{-9}	9. -100000
5. 375×10^{-3}	10. 1.3

1) Convert the ten decimal numbers of List 1 to standardised floating point form, giving the result as the number pair (mantissa, exponent), where
mantissa = sign bit. octal fraction
exponent = decimal integer.

For example, 4.5 = (0.44, 1).

2) Store the first five numbers of List 1 in standardised floating point form in full words iA1, i = 0 to 4, using only B-type instructions.

List 2

```
* = 0

1)  +1(:+2)           7)  +1(2)
2)  +K765(:-2)        8)  -K2.3(2):K10
3)  -K1.1:5           9)  4(:-2)
4)  +0:26            10)  -2(:3):4
5)  -1(:-1):2
6)  +0
```

3) Write in octal form the numbers stored in full words A1 to A10 by the ABL program shown in List 2. For Example, (A6) = J4.

6.2 Set II (see chap. 3)

Exhibit the contents of the accumulator after the execution of each of the following sets of instructions. Use the notation

ay' = J(3 octal digits)

m' = octal digit . (13 octal digits, omitting right hand zeros)

l' = . (13 octal digits, omitting right hand zeros)

sign l' = 0 or 1.

For example, the instruction

```
1) + 16
324      0      0      A1
```

leaves A in the state

ay' = J002

m' = 0.2

l' = .0

sign l' = 0.

1)
```
1) -16
324      0      0      A1
```

2)
```
1)+K21.5 : 13
334      0      0      A1
```

3)
```
1) +K275.123456               0:26
324      0      0      A1
320      0      0      1A1
```

4)
```
1) +K73:3
334      0      0      A1
340      0      0      J4
```

5) 1) -10

344	0	0	A1
101	124	0	A1
124	124	124	0

In problems 6) onward, in addition to showing the final state of the accumulator, indicate accumulator overflow (AO), exponent overflow (EO) and exponent underflow (EU) and instruction(s) which caused them.

6) 1) -1

346	0	0	J4
331	0	0	A1

7) 1) -1(:-127)

344	0	0	A1
121	124	0	0
355	0	0	J4

8) 1) +K2 -K3

324	0	0	A1
352	0	0	1A1

9) 1) -K1(:K177)

325	0	0	A1

10) 1) +5 2)+1(:-24)

324	0	0	A1
320	0	0	A2

11) 1) +K.4

324	0	0	A1
364	0	0	J4
365	0	0	J4

12) 1) -1(:64)

314	0	0	A1
372	0	0	A1
365	0	0	J4

13) 1) +.125(:-127) +0

324	0	0	A1
332	0	0	1A1
366	0	0	J4

14) In this problem exhibit the contents of the store line before and after the instruction is performed

1) -1

335	0	0	A1
356	0	0	A1

15) 1) +1 2) -1

324	0	0	A1
373	0	0	A2

6.3 Set III (see chaps. 3 and 6)

In questions 1), 2) and 3) the standardized floating point numbers a_0, a_1, ..., a_{99} are held in addresses iA79, i = 0 to 99.

1) Place their double length floating point sum in the accumulator, round to single length and store the result in A80.

2) Place the integral part of a_0 in the accumulator in double length fixed point form (25, int.pt.a_0 $\times 8^{-25}$).

3) Assume that the accumulator holds a double length floating point number Z. Compute the product of that number, double length, with v = (A90) and leave the result in the accumulator.

In questions 4), 5), 6) and 7) the integers x_0, x_1, ..., x_{99} are stored in (iA100), i = 0 to 99, in the substandard floating point form (12, $x_i \times 8^{-12}$).

4) Working fixed point, store their alternating sum: $x_0 - x_1 + x_2 - \ldots$ in A99.

5) Place the integral part of the quotient x_0 / x_1 in A399 and the remainder in 1A399, both in the substandard form. Assume x_0 and x_1 to be positive.

6) Store $x_3 \times x_4$ as a standardised double length number in A341,1A341.

7) Working fixed point, form the quotient 936.29 / 48.01 to 3 octal places and store it in A309 in the substandard form (10 , x). Store the remainder in standardised floating point form in 1A309. The operands should be stored by program.

7. EXTRACODE FUNCTIONS (see chap. 7)

1) The double length no. x is held in location 50:
The double length no. y is held in location 60:
The double length no. z is held in location 70:
The double length no. X is held in location 10:
The double length no. Y is held in location 12:

a. Form $z - (z/x)y + z/X + z/Y$
b. Form $|x| + |y| + |z|$ in the accumulator.

2) Two 3 × 3 matrices A and B are stored in the following way:

	A	B
1st row vector is stored in	10 11 12	40 41 42
2nd row vector is stored in	20 21 22	50 51 52
3rd row vector is stored in	30 31 32	60 61 62

Replace matrix A by matrix A + B.

3) The complex numbers A and B are stored as follows

$A = u + iv$ is stored in 50:
$B = x + iy$ is stored in 60:

The complex accumulator is taken as 70:

a. Form $A(B)^{\frac{1}{2}}$ in the complex accumulator
d. Form $(1/B) + \exp. A$ in 50:

4) Write a routine to do the following in the order given:

a. Read 160 characters from input 5 into the least significant 6 bits of halfwords starting at location 10.

b. Pack these characters in order into full words starting at location 100.

c. Copy the first two characters of each word separately in order, into B registers B1 to B40.

d. Check that each character has been correctly set in the appropriate B register. Use extracode 1204.

5) A list of 10 numbers is held in store locations 20 . Using extracode 1436 find the standard deviation of these numbers, leaving the result in the accumulator.

6) a. Write a routine R3 to shift the logical accumulator circularly to the left by n character positions. Assume that n is stored in B79 upon entry and use B80 for the link.

b. Working in the logical accumulator arrange that the floating point number in location 500 has a zero sign bit.

8. <u>INPUT AND OUTPUT OPERATIONS</u> (see chap. 8)

LIBRARY ROUTINES L1 and L100 should be used and only questions 9 and 11 should be treated as complete programs.

1) Read a floating point number from input stream 1 and output its modulus in fixed point form, on a new line, on output stream 1. The number is less than 100 and is required to two decimal places.

2) Read ten mixed numbers from input stream one to the store locations starting at word 1000.

3) Modify the solution to example 2 so that n numbers are read, where n is an integer at the head of the data.

4) Output the floating point numbers stored in locations 550 to 577 inclusive, in floating point form to 7 significant figures, each number on a new line. Use output two.

5) Search the information on input stream 2 ignoring everything until the letter X (internal code 7.0) is encountered. Then read the following number to the accumulator and arrange that any further information on this line is ignored.

6) Output the 20 numbers stored in word 512 onwards, five per line, in signed fixed point form, separated by at least two spaces. The numbers are less than 1000 and are required to four decimal places. Use output stream one.

7) Input stream 1 consists of a set of numbers of the order 10^5 terminated by -1 (which does not form part of the set). Form the sum of these numbers and print it, preceded by the number of numbers read. End output on this line.

8) How could the solution to example 7 be modified to accept *(internal code 1.6) as the terminator? Set any parameters required by L100.

9) Read a set of 10 numbers from input stream 6. For each number x, output x on a new line, followed by $\sqrt{x}$. If x is negative, replace $\sqrt{x}$ by the letter I (internal code 5.1). Output should be in floating point form to five significant figures. Arrange to separate the exponent from the mantissa by a comma instead of enclosing it in brackets.

10) Write a subroutine, R2, to evaluate sine θ and cosine θ given θ (in degrees) in the accumulator on entry. Store the functions in A99 and 1A99 respectively. (Use B50 for the link and make use of the extracodes 1730, am' = sin s and 1732, am' = cos s where s must be in radians. π = 3.141593).

11) Use this routine to tabulate sine θ and cosine θ for θ at 10° intervals between 0° and 180°. Print θ as a three digit unsigned integer and sine θ , cosine θ as five digit fractions preceded by + or -. Ensure there are two spaces between each number and use output three.

12) Print, on output zero, the character ≠. Output zero is a 7 hole paper tape punch. Write out the contents of B81 in binary just before exit to L1.

13. Store the text:-

TAPE FAULT Ø ≠ 7

in successive character locations in the store. Put the address of the first character in B21 and the number of characters in B22.

14. In question 11, assuming output to be to 7 hole paper tape, arrange for the columns to be headed θ, sin θ, cos θ respectively.

15. Adapt the program of question 9 to read a title of one line from immediately before the first x on the data tape and to print it, followed by 13 new lines, after the last $\sqrt{x}$.

For the following examples use input and output extracodes.

16. If input 5 is at the end of the current record read the first character of the next record; if not read the next character. Store it in the least significant 6 bits of ½ word A10 if it is a six bit character or A11 and 0.1 A11 if it is a binary character.

17. Delete all previous output 4. Write the binary representation of M on 7 hole paper tape to Output 4, end the record and arrange for printing as soon as possible.

18. Read a record (not binary) from input 3 and write it to output 5 unless there are more than 12 characters. In this case write the first 12 only and do not end the record. What alterations would be necessary to end the record in the latter case with 5 new lines?

9. MAGNETIC TAPE OPERATIONS (see chap. 9)

9.1 Fixed Length Transfer

1) Transfer 5 blocks of information from tape 2 beginning at block 1 into store blocks 10 onwards.

2) Read blocks 5, 4 and 3 of tape 1 to store blocks 3, 2 and 1 respectively.

3) Four blocks of information are stored on tape 1 in blocks 1 onwards. Transfer each half word into a separate full word location in the store starting from store location 100.

4) Information is stored in blocks 1-5 inclusive in the store. Write blocks 1 and 2 into blocks 4 and 5 of tape 1, block 3 into block 7 of tape 1 and blocks 4 and 5 into blocks 1 and 2 of tape 1 respectively. The write operation should be performed in the order given.

5) Read block 5 of tape 1 into store locations starting at 1: write this block back to the same block on tape 1 but reverse the order of the words in the block.

6) Write 10 blocks from A1 to the 10 sections from that whose number is held in B1, on the tape whose number is held in B2 as a half word address (use substitution register B121).

7) Mount a tape whose title is MERGER INC. TAPE 2AB and assign the number 2 to it. (Assume that the title has also been stored in exactly one section other than section 0 by a previous program, using the ABL C directive and storing the title beginning with word 0 of the block in question). Station the tape just before the section other than 0 holding this title in a manner which will keep the tape moving forward as fast as possible.

8) Write a complete program including JOB description to read 4 blocks of numbers, starting at block 1, from a TAPE having title XYZ - 3 into blocks 3 - 6 of the store. Find the mean value of the number in each block, write the blocks back to the tape having title XYZ - 4 into blocks 1 - 4 write into block 5 of the same tape the mean values of the separate blocks of numbers in the correct order. The contents of the remainder of block 5 are immaterial.

9) Tapes 1 and 2 are stationed at unknown positions P_1 and P_2. Reposition the tapes at sections $|(P_1 + P_2)/2|$'. (The square brackets indicate that the integral part of the expression is required). Transfer 100 blocks from tape 1 sections $|(P_1 + P_2)/2| - i$ where i goes from 1 to 100 to tape ? sections $|(P_1 + P_2)/2| + j$ where j goes from 0 to 99.

9.2 <u>Variable Length Transfers</u>

1) Search for word 173 of section 7 of tape 6.

2) Using a buffer in blocks six and seven of the store write 750 words from A3 to the beginning of tape two, preceded by a '7' marker and followed by a '1' marker.

3) As 2) but precede information by a '3' marker and follow it by a '4' marker.

4) a. Search for word 6, section 4 of tape 3, where a marker is located.
   b. Set up a buffer for reading from tape 3 in the first available block after A6.
   c. Read 100 words to A1. Assume first string has more than 100 words.
   d. Read to A2 until a marker of order $\geqslant 3$ is encountered.
   e. Read an equal number of words to A3.
   f. Read not more than 200 words to A4, stopping if a marker $\geqslant 4$ is encountered.

5) Place a '3' marker at word 3 of block 100 of tape 4 without any further alteration of the tape. A marker already exists at this point.

6) a. Transfer the first and fourth strings of information on tape 3 to consecutive store locations beginning at A2 (set up a two block buffer at some convenient point).

   b. Release the buffer.

7) Read the 100 words preceding the present position of tape 6 to store locations 216 to 315.

8) Read the first 700 words from tape one which has been written by fixed block transfers.

9) a. Read the first string from tape one.

   b. Write 160 words from A7 onwards to tape one beginning at the fourth word of the hundredth section, and ending with a '3' marker.

   c. Read the second string from tape one.

10) Write a subroutine to read a string from the present position of tape four, assuming other tapes are in use in the main program. The buffer has been set up by the main program.

11) Write 30 words from A7 onwards to tape one ending with a '1' marker, unless it is engaged in fixed block transfers.

12) Read exactly half of the first string of tape six to A1 onwards (assume the string consists of an even number of words).

10. <u>JOB DOCUMENTS</u> (see chap.).

1) Write the job description, headings, titles and end of tape markers of all other documents, for the following programs. All documents are on separate tapes (or stacks of cards) unless otherwise stated.

   a. Job title 'MEAN CALCULATION'.
   Program in A.B.L. on same tape as job description.
   One data tape called 'X3 RESULTS'.
   One output stream to paper tape or line printer, of 3,000 characters.
   Estimated computing time 3 secs.
   Store used 6 blocks.

b. Job title 'MONTE CARLO RALLY 1962'.
Program in Extended Mercury Autocode 'RALLY ANALYSIS'.
Two data tapes 'RESULTS 1962'
'WEATHER CONDITIONS'.
Two output streams:- one block to punched cards four blocks to line printer.
The following magnetic tapes need to be mounted at the start of the program:-
Single tape: 'ENGINE SPECIFICATIONS 1962'.
A free tape to be given the title 'RALLY RESULTS 1962-1967'.
The first tape of a file consisting of two tapes called 'RALLY RESULTS 1950-1955'
and 'RALLY RESULTS 1956-1961'.
Four magnetic tapes may be in operation at any one time.
Estimated time 10 mins. plus 5 mins. tape waiting time.
Store used 25 blocks.

c. Job title 'F/126/JKB 8 FEB. 63'.
Program 'F/126/JKB' in ABL.
Two data tapes, one combined with job description and one called 'SPECIAL RANDOM NUMBERS'.
One output stream of 20 blocks to be stored on a free, private magnetic tape called '126-CLASSIFIED RESULTS'.
This is to be printed later on 5 hole paper tape.
Time 25 seconds. Store required 3 blocks.

d. Job title 'SURVEY ANALYSIS 14'.
Program 'SURVEY ANALYSIS' in ABL.
One data document on two separate paper tapes, title 'SURVEY RESULTS'.
One output stream to 7 hole paper tape of 5000 characters.
Time 20 seconds. Store required 20 blocks.

e. After d. the program is stored at system magnetic tape address S5/3/500.
Run the program again with job title 'SURVEY ANALYSIS 15', using data document 'SURVEY RESULTS/2'.

2) a. Store all data for 1)b. on a free magnetic tape which is to be given the title 'DATA FOR MONTE 62'.
Assume the document 'RESULTS 1962' takes less than 100 sections.

b. Write a job description to use this magnetic tape in place of the two data tapes.

c. Write a steering tape to print the results of job 1)c.

d. What changes would be necessary in 2)a. and 2)b. if the two data documents were on one tape separated by a ***C marker?
Assume the following information is printed following the transfer of data to magnetic tape:-

```
DATA FOR MONTE '62
RESULTS 1962 1/0          61
WEATHER CONDITIONS 61/413 32
```

3) Write a complete program including job description etc., to read 10 numbers from a data tape and find, and print, their mean.

11. ADVANCED PROGRAMMING TECHNIQUES (see chap. 11).

11.1 Store Extracode Operations

1) Read block A1 to the operands side of the core store, duplicate it as A2 on the drum, rename block A1 as A3. Release A3 to the drum and lose A2.

2) Achieve the same final result as 1) by any more efficient method.

3) A list of numbers extends from location A1 to location A2. A1 is the first word of a block and A2 is the last word of another, higher numbered, block. The list may be of considerable length. It is required to find the sum of the products of pairs of consecutive numbers in the list, and once this is done the list will not be needed again. Program this efficiently. The final result is to be left in location A10 and the possibility of exponent overflow can be disregarded. Set A1, A2 and A10 explicitly for a list containing 512 × P1 numbers.

4) 100 blocks of main store labelled consecutively A10, A10+1:, ... A10+99: are processed one at a time starting with A10 and working up to A10+99:. The processing takes about 10 milliseconds and the results for any block, A10+n say, must go to block A11+n, for all n. The original block is not wanted again after A11+n has been filled. Program the process efficiently for a block allocation of 102.

5) A bad programmer has abandoned a routine for evaluating sin α for values of α from 0° to 90° with an interval between adjacent values of 0.0001757848680°. He has written all these values (in radians) sequentially into 100 blocks of the main store starting at location 10:0. He wishes to overwrite these arguments with sin α. Can you finish his routine using the original list of α's and the store extracodes? The link is to be in B40.

11.2 Branching

1) A magnetic tape with the title 'STORE 3:SALES RECORDS' contains records of the sales of various types of goods stocked by one store of a chain. Each type of article has one 512 word section to itself. The first half-word of each section is 3, which is a code number for the store, and the second half word is a code word for the type of article.

   A second tape, called 'AMENDMENTS TO SALES RECORDS/ MARCH' contains corrections of sales information for all stores of the chain, one section being allocated per store for each type of article for which alteration is necessary. The same convention as the first tape is used for the first half word of each section, which varies according to the store concerned. The remainder of the section, however, is not in the required form and must be processed by a subroutine R5 whose specifications are given below. The second half word of the processed section will be the same as that of the block it has to replace.

   The sections for one type of article on the amendments tape are grouped together, the order of types being the same as that of the sales record tape. The last useful section of the amendments tape is followed by a section with zero in the first half-word.

   Write a program, using three branches, to mount the tapes on different channels and amend the necessary sections of the first tape.

   Specifications of R5

   a. Link in B90
   b. Operates on one block of information, beginning at 2:
   c. Produces a block suitable for transfer to sales tape in block 3:
   d. Entry point at first instruction
   e. Uses B-registers B10 to B20

11.3 Monitoring and Trapping

1) Arrange that the program jumps to the routine beginning at location A3 when the last item has been read from a data tape. If computer failure should occur re-enter the program at the point where the fault occurred, after recovering the contents of the extracode working registers.

2) Arrange that, if a fault other than a computer or tape failure is encountered, the contents of store locations A1+1, A1+2 and A1+3 will be printed out in floating point form to 10 significant digits on

separate lines, by a private monitor routine which will be entered after printing the type of fault. After the private monitor, terminate the program. In the case of computer failure arrange to repeat the instruction on which the fault occurred and continue the program.

3) Write a routine to determine the first integer K, greater than 2, for which sin K > 1-1/K, limiting the routine to no more than 5 seconds by means of the local instruction counter. If the search is successful print the result as an integer on output 0.

4) Write 20 consecutive blocks, starting at block 40: to tape 4, beginning at the present position of the tape (assuming that this is at the beginning of a block). If tape 4 holds all the blocks, end the program; otherwise, place the remaining blocks on a 'free tape' which is allocated number 5 and title OVERFLOW BLOCKS. Print on output 0 the number of blocks which overflowed and end the program.

5) Information is stored on tape 2 starting at the 1st word of block n (n is stored in B2). Every 10th block, beginning with the first, must be read into store block 10: and then processed by routine R4. Arrange to enter routine R5 if an illegal search is made or if the end of the tape is reached. Also arrange that the program is restarted completely in the event of a tape failure.

6) Transfer the information in store blocks 1: to 15: to tape 99, establishing a restart point so that if a tape failure occurs the tape will be re-positioned at the beginning of the current block and the transfer continued from there.

12. COMPLETE PROGRAMS

12.1 Example using Fixed Length Tape Transfers

1,000 blocks of information are stored at the beginning of a tape called 'RESULTS OF EXPERIMENTS R35'. This information is to be processed by the computer, each block of information producing one block of results which are to be written to a free tape.

The processing is done by a subroutine R3, entered at the first instruction, which requires a link in B90. In processing block n, R3 uses the results of processing block (n-1) and it requires block n to begin at A3 and the results from (n-1) to begin at (A3-1:). For processing block 1, block (A3-1:) must be filled with floating point zeros. The results produced will overwrite the information in block A3.

Processing one block will take approximately 1/10th of a second.

Sketch a flow diagram which will use the tapes as efficiently as possible and draw up the complete program document required, leaving a space for R3.

12.2 Example using Variable Length Tape Transfers

A magnetic tape with title 'F123/JKB/B6 20/1/63' contains information written in variable length strings. The first half word of each string is a keyword to identify the information which follows.

It is necessary to replace some of these strings by amendment strings which are on a magnetic tape called 'F123/JKB/B6 AMENDMENTS 20/2/63'. The keyword of each string on this tape corresponds to that of the string which it must replace. There is no guarantee that the amendment string is the same length as the string which it has to replace.

All strings are less than 100 words in length and are separated by a '1' marker. The information on each tape begins and ends with a '2' marker.

Write a complete program to carry out the amending leaving the final result on a tpae to be entitled 'F123/JKB/B6 20/2/63', and draw up the necessary job description.

13. SOLUTIONS

13.1 Solutions to Modification, Counting and Testing Exercises Set I (4.1).

```
1.      121     1       0       99
      1)113     1       1       A4
        203     127     1       A1

2.    1)+1
        324     0       0       A1
        170     0       0       P26
        224     127     0       A2
        123     1       0       P26-1
      3)362     0       1       A10+P26-1
        201     127     1       A3
      2)- - - - - - - - - - - - - -

3.a)    121     3       0       0
        113     0       0       A49
   ┌→1) 101     2       3       A17
   │    102     2       3       A17+0.4
   │    114     2       0       A49
   │    172     3       0       P17+13
   └─── 221     127     3       A1

  b)    121     6       0       A17
      1)101     2       6       0
        102     2       6       0.4
        214     127     2       A2
        124     6       0       1
        121     127     0       A1
      2)- - - - - - - - - - - - - -

4.      121     2       0       0
        346     0       0       J4
   ┌→1) 320     0       2       60
   │    124     2       0       6
   │    172     2       0       60
   └─── 225     127     0       A1

5.      121     3       0       0
        121     4       0       1D1
   ┌→1) 124     3       4       0
   │    113     0       3       0
   │    172     3       0       4950 D1
   └─── 220     127     4       A1

6.      1312    5       5       0

        121     7       0       P7-1
   ┌→1) 1312    6       6       0           b6' = b6².
   └─── 203     127     7       A1

        1314    1       2       0           b1' = int pt. (b1/b2)
        124     1       0       1
        113     1       0       A6
```

```
7.        113      0       0       100
          121      2       0       0
   ┌→1)101      3       2       A6
   │      121      4       2       0
   │┌→2)1312     3       12      0
   │└── 203      127     4       A2
   │      114      3       0       100
   │      172      2       0       P41
   └──── 221      127     2       A1

8.        121      1       0       P13
          121      4       0       A90
          121      30      0       0
          121      3       0       -0.4
   ┌→2)124      3       0       0.4
   │      150      1       3       A77
   │┌── 226      127     0       A1
   ││     124      30      0       1
   │└→1)172      4       3       A77
   └──── 224      127     0       A2

9.        121      90      0       A13
    ┌→1)101      1       90      0
    │     172      1       90      0
    └─── 221      127     90      A1

10. 10)+0
          121      80      0       -99D1
    ┌→3)101      1       80      A1+99D1
    │     1524     0       1       0          am' = n, l' = 0
    │     101      1       80      A2+99D1
    │     1574     0       1       0          am' = ai/bi
    │     320      0       0       A10
    │     356      0       0       A10
    └─── 200      127     80      A3
```

Set II (4.2)

Group A

```
1.    1)+0
          324      0       0       A1
          121      1       0       -999
       4)356      0       1       999A3
          201      127     1       A4

2.    1)+0
          324      0       0       A1
          121      1       0       -1P7
       2)356      12      1       0
          203      127     1       A2
```

3.	3)+0			
	324	0	0	A3
	121	1	0	9D1
	→5) 101	2	1	A1
	122	2	0	1
	101	3	1	A2
	→4) 356	2	3	0
	203	127	2	A4
	202	127	1	A5

4.	121	3	0	1
	121	2	0	-0.4
	121	1	0	-999
	→3) 124	2	0	0.4
	152	3	2	A7
	225	127	0	A3
	113	0	2	A7
	201	127	1	A3

Group B

1.	121	1	0	0
	121	2	7	0
	→1) 104	1	2	0
	101	2	2	0.4
	172	2	7	0
	225	127	0	A1
	113	1	0	A3

2.	121	1	0	0
	121	2	0	A13
	→1) 101	3	2	0
	113	3	1	A12
	101	2	2	0.4
	172	2	0	A13
	221	127	0	A1

Group C

1.	→ 324	0	0	A1
	356	0	0	A40
	121	1	0	P7
	→4) 121	2	1	0
	→3) 121	3	1	0
	122	3	2	0
	324	0	2	A1
	362	0	3	A2
	320	0	0	A40
	356	0	0	A40
	203	127	2	A3
	203	127	1	A4

```
2.  ──→ 121     1      0      P7
        324     0      1      A1
        320     0      1      A2
        356     0      0      A3
   ┌─4) 122     1      0      1
   │    324     0      1      A2
   │    374     0      0      A3
   │    320     0      1      A1
   │    356     0      0      A3
   └──→ 215     127    1      A4
```

13.2 Solutions to Shifts and Logical Operations on the Contents of B-registers.
Set I (5.1).

```
1.a)    K27.0
        J7777751

  b)    1:1
        3:464

2.a)    J7370000.7

  b)    J07737

3.a)    126     7      0      J7777777.7

  b)    121     7      9      0
        126     7      10     0

4.      165     8      9      K360.0
        1343    8      0      3          circular shift of b8.
        127     6      0      J77741
        167     6      8      0

5.      121     1      0      30
        126     1      31     0

        121     2      0      30
        127     2      31     0

6.      1344    7      0      23         logical shift of b7.
        1345    7      0      3

7.a)    127     71     0      J3777777.7

  b)    167     71     0      J4

8.      1344    3      0      9
        164     3      4      K77.7
        1344    3      0      3
        127     4      0      J77777
        1345    4      0      3
        167     4      3      0
```

9.	1342	80	0	18
	127	80	0	K7777777.6
	165	2	13	K.1
	167	80	2	0
	1343	80	0	18

10.	121	1	0	0.1
	121	3	0	0
	121	4	0	-24
→1)	214	4	0	A2
	124	4	0	1
	163	1	0	0
	121	2	1	0
	127	2	60	0
	214	127	2	A1
	126	3	0	1
	214	127	3	A1
	126	60	1	0
	121	127	0	A1
2)				

11.a)	127	12	0	K777777.0
	165	2	13	J7K.7
	167	12	2	0
b)	1344	12	0	12
	165	2	13	K777.7
	1345	2	0	12
	167	12	2	0
c)	127	12	0	K707070.7
	165	2	13	J7070707
	167	12	2	0

Set II (5.2)

Group A

1. →	121	1	0	0
	121	2	0	K.1
→	163	2	0	0
	211	127	62	2*
	167	1	2	0
→	163	62	0	0
	211	127	2	-4*
	121	62	1	0

2. The following solution is based on the fact that if $a \neq 0$ then $a \& (a - 1)$ has exactly one fewer 1 bit than a.

	121	31	0	0
	214	127	17	3*
→	124	31	0	1
	127	17	17	-.1
→	215	127	17	-2*

```
4.        121      3       0       -499.4
   →1)101      2       3       A77 + 499.4
          106      2       3       A29 + 499.4
          215    127       2       A2 ─┐
        ─ 200    127       3       A1  │
          121    127       0       A4  │
       2)113      2       3       A29 ←┘
        ─ 200    127       3       A1
       4)
```

Group B

```
2.        121      1       0       -23
          121      2       0       K0.1
          163      2       0       0
          113      2       1       23A40
          203    127       1       -2*

3.        121     80       0       0
          121    121       0       48D1
          121     60       0       -1
       →1)124     60       0       1
          1344   122      60       0
          121    122      61       0
        ─ 211    127      61       2*
          124     80       0       1
        → 172    121       0       25D1
        ─ 222    127     121       A1
          127     80       0       1

4.        1)  (working space)
          121    121       0       25D1
          123     60       0       1
       3)124      60       0       1 ←
          113    122       0       A1
          101     80       0       A1
          123     61       0       23
      →2)1345     80      61       23
          121     81      80       0
          127     81       0       J4
          1344    81      60       0
          1347    81      61       23A40
        ─ 201    127      61       A2
          172    121       0       48D1
          220    127     121       A3
```

5. The solution may be obtained by placing G^t in the H position as above and then performing the following instructions.

	121	121	0	1D1
5)	121	60	0	23
4)	113	122	0	A1
	101	70	60	A40
	107	70	0	A1
	121	71	0	0
	214	127	70	A6
	124	71	0	0.1
	127	70	70	-0.1
	215	127	70	-2*
	124	121	0	24D1
	210	127	71	4*
	143	122	0	0
	127	122	0	-(J4+K0.1)
	121	127	0	2*
6)	167	122	0	J4
	123	121	0	23D1
	203	127	60	A4
	172	121	0	25D1
	225	127	0	A5

The elements of the matrix F × G are computed row by row, the elements of each row being computed in reverse order and shifted in from the left. The element a_{ij} is computed as the number of ones, taken modulo 2, in the quantity b(i+1) & (jA).

13.3 Solutions to Accumulator Operations. Set I (6.1).

1.a) (0.0, -128)

b) (1.634, 3)

c) (0,1, 0)

d) (1.0, -3)

e) (0.3, 0)

2.a)

121	1	0	J4
113	1	0	A1
113	0	0	.4A1

b)

121	1	0	J007634
113	1	0	1A1
113	0	0	1.4A1

c)

121	1	0	1U9
113	1	0	2A1
113	0	0	2.4A1

d)	121	1	0	J773
	113	1	0	3A1
	113	0	0	3.4A1
e)	121	1	0	K3U9
	113	1	0	4A1
	113	0	0	4.4A1

3.a) (A1) = J0061

b) (A2) = J002765

c) (A3) = J013777767

d) (A4) = J064

e) (A5) = J005777

Set II (6.2).

1. ay' = J002

m' = 7.6

l' = .0, sign l' = 0

2. ay' = J015

m' = 0.0...,021 (13 octal place fraction)

l' = .0, sign l' = 0

3. ay' = J003

m' = 0.275 (integral part of (A1))

l' = .0, sign l' = 0

4. ay' = J012

m' = 0.73

l' = 0., sign l' = 0

5. ay' = J012

m' = m

l' = .66, sign l' = 1

6. ay' = J0 — A0 set by 330

m' = 1.0

l' = .0

sign l' = 0

7. ay' = J763

m' = 7.0

l' = 0

sign l' = 0

8. ay' = J002

m' = 7.72

l' = 0

sign l' = 1

9. ay' = J200

m' = 0.1

l' = .0

sign l' = 0

10. ay' = J001

m' = 0.5000000000001

l' = .000000000001

sign l' = 0

11. ay' = 0

m' = 7.4

l' = 0

sign l' = 0

12. 372 0 0 A1 gives E0

ay' = J200

m' = 0.1

l' = 0

sign l' = 0

13. ay' = J601

m' = .1

l' = 0

sign l' = 0

14. The contents of the store line before and after the instruction is the same.

i.e. oo0010.....0

and the 335 instruction sets A0.

15. ay' = J001

m' = 0.1

l' = 0

sign l' = 0

Set III (6.3)

1. 1)+0 (working space)
2)+0 (working space)

	121	1	0	-97	count
	121	2	0	13U13	to modify exponent
	324	0	0	A79	a_0 in
	300	0	0	1A79	add a,
3)	357	0	0	A1	store right half
	110	2	0	A1	with correct exponent
	300	0	1	99A79	add in next a
	356	0	0	A2	store left half
	355	0	0	J4	standardize right half
	320	0	0	A1	add original right half
	300	0	0	A2	add in left half
	201	127	1	A3	repeat
	361	0	0	J4	round result
	356	0	0	A80	store

2. 1)0:26

324	0	0	A79
330	0	0	A1
364	0	0	J4

3. Let z_1 and z_2 donate the more and less significant halves of Z, and w_1, w_2 the halves of the products VZ. Thus VZ = $w_1 + w_2$.
Working space: A1, A2 ... .

356	0	0	A1	store Z_1
355	0	0	J4	standardise Z_2
342	0	0	A90	form VZ_2
356	0	0	A2	store $(VZ_2)_1$
355	0	0	J4	std. $(VZ_2)_2$
356	0	0	A3	store $(VZ_2)_2$
324	0	0	A1	load Z_1
342	0	0	A90	form VZ_1
356	0	0	A4	store $(VZ_1)_1$
355	0	0	J4	std. $(VZ_1)_1$
320	0	0	A3	add $(VZ_2)_2$, R
356	0	0	A3	store first partial W_2
324	0	0	A4	load $(VZ_1)_1$
300	0	0	A2	add $(VZ_2)_1$
356	0	0	A2	store partial W_1
355	0	0	J4	std. second partial W_2
300	0	0	A3	add first partial W_2
300	0	0	A2	add partial W_1

4.

346	0	0	J4
121	1	0	-98
330	0	1	98A100
331	0	1	99A100
201	127	1	1*
201	127	1	-3*

5.

345	0	0	A100
364	0	0	J4
375	0	0	1A100
121	124	0	12U12
357	0	0	A399
356	0	0	1A399

6.

324	0	0	3A399
342	0	0	4A399
356	0	0	A341
355	0	0	J4
356	0	0	1A341

7.

```
1)936.29 : 10
2)48.01  : 10
345    0      0      A1
364    0      0      J4  )
364    0      0      J4  )  position dividend
364    0      0      J4  )
375    0      0      A2
121    124    0      10U12
347    0      0      A309
121    124    0      7U12
340    0      0      J4
356    0      0      1A309
```

13.4 Solutions to Extracode Functions (7.)

1.a)

```
1504   0   0   0    a' =
1576   0   0   50   a' = z/x
1543   0   0   60   a' = -yz/x
1500   0   0   70   a' = z-yz/x
1556   0   0   16   (16:)' = a
1504   0   0   70   a' = z
1576   0   0   10   a' = z/X
1556   0   0   18   (18:)' = a
1504   0   0   70   a' = z
1576   0   0   12   a' = z/Y
1500   0   0   16   a' = z/Y + z - yz/X
1500   0   0   18   a' = z/X +z/Y + z - yz/x
```

b)

```
1567   0   0   50   a' = |x|
1556   0   0   50   (50:)' = |x|
1567   0   0   60   a' = |y|
1556   0   0   60   (60:)' = |y|
1567   0   0   70   a' = |z|
1500   0   0   50   a' = |z| + |x|
1500   0   0   60   a' = |z| + |x| + |y|
```

2.

```
   121    3     0   -2    Count in B3
   121    4     0   10    Modifier in B4
   121    1     0   0     Clear B1
   121    2     0   30    Modifier in B2
1) 124    1     4   0     Locate A row
   124    2     4   0     Locate B row
   1430   1     0   3     Add
   201    127   3   A1    Count and repeat
```

3.a)

```
121    1   0   70   b1' = 70
1410   1   0   60   ca' = (60:)^½
1462   1   0   50   ca' = ca. (50:)
```

b)

```
121    1   0   70
1414   1   0   60   ca' = 1/(60:)
1456   1   0   60   (60:)' = ca
1402   1   0   50   ca' = exp(50:)
1420   1   0   60   ca' = ca + (60:)
1456   1   0   50   (50:)' = ca
```

```
4.a)    1050    0     0     5          Call input 5
         121    41    0     -159D1     Count-modifier in B41
      1)1101    90    0     A3/L100    Read next character
         113    81    41    10+159D1   Count and repeat

  b)     121    41    0     100        Locate full words
         121    42    0     10         Locate characters
        1253    41    0     160        Pack characters

  c)     121    41    0     -19.1      Modifier in B41
         121    121   0     0          Clear B121
         124    121   0     1D1        Set B121 as next B-line
        1250    122   41    119.1      Write character to B-line
         172    121   0     40D1       Count B-lines
         225    127   0     2*
        1117    0     0     0          End
         211    127   41    3*         Modifier even ? Jump
         124    41    0     0.1        Add 0.1 to odd modifier
         121    127   0     A1         Continue
         124    41    0     0.7        Add 0.7 to even modifier
         121    127   0     A1         Continue

  d)     121    41    0     -1         Modifier in B41
         121    121   0     1D1        Point at B1
      1)1251    122   0     200        Next character to 200
         124    121   0     1D1        Advance pointer
        1251    122   0     200.1      Next character to 200.1
         113    121   0     201        Preserve B121
         124    41    0     1          Advance modifier
        1601    0     41    100        Next pair of characters
                                         to G
        1204    42    0     200 ]      Compare and fault
         172    42    0     2   ]      unless characters
         227    127   0     A4  ]      agree
         101    121   0     201        Restore B121
         170    121   0     40D1]      Repeat, advancing
         220    127   121   A1  ]      pointer


5.       121    1     0     9          b1' = 9
         346    0     0     J4         am' = 0
      1)320     0     1     20         am' = am + (20 = b1)
         203    127   1     A1         conditional jump
        1574    0     0     10         am' = am/10
         121    8     0     9          b8' = 9
      2)356     0     8     30         (30 + b8)' = am
         203    127   8     A2         conditional jump
         121    5     0     20         Establish x vector
                                                  _
         121    6     0     30             "     x   "
        1431    5     0     10         Compute x-x vector
         121    4     0     10         Establish working space
        1434    4     0     10         store copy of x-x
        1436    4     0     10         Compute sum of squares
        1574    0     0     10         Compute s.d.
```

6.a)

	R3				
	122	79	0	1	Count in B79
	121	1	0	0	Clear B1
	1265	1	1	0	Shift upper char. to B1
	124	99	1	0	Add it in at bottom
	203	127	79	-2*	Count and repeat
	121	127	80	0	Link

b)

	1)-1				
	1601	0	0	500	g' = (500)
	1630	0	0	A1	Make m.s. bit of g zero
	1613	0	0	500	(500)' = g

13.5 Solutions to input and output operations (8.)

1.

	1050	0	0	1	Select input 1
	121	90	0	A1	Set link in B90
	121	127	0	A1/L100	Read next number to accumulator
1)	121	90	0	A2	Set link in B90
	1060	0	0	1	Select output 1
	121	89	0	2:2.6	Set style for accumulator print
	121	127	0	A1/L1	Print accumulator
2)	- - - Next Instruction				

2.

	1050	0	0	1	Select input 1
	121	3	0	-9	Set count in B3
→5)	1101	90	0	A1/L100	Read next number to acc.
	356	0	3	1009	Store number
	201	127	3	A5	Count and repeat

3.

	1050	0	0	1	Select input 1
	1101	90	0	A2/L100	Read n to B81
	121	2	81	-1	b2' = n-1
	123	3	2	0	Count-modifier in B3
→5)	1101	90	0	A1/L100	Read next number
	356	2	3	1000	Store number
	201	127	3	A5	Count and repeat

4.

	1060	0	0	1	Select output 1
	121	1	0	-27	Count in B1
	121	89	0	1:6.3	Style in B89
	121	90	0	3*	Set link
→1)	324	0	1	577	Next number to accumulator
	121	127	0	A1/L1	Print accumulator
	201	127	1	A1	Count and repeat

5.

	1050	0	0	2	Select input 2
→2)	1101	90	0	A3/L100	Read next character to B81
	122	81	0	7.0	)Repeat unless character
	215	127	81	A2	)is "X"
	1101	90	0	A1/L100	Read next number to acc.
	1101	90	0	A4/L100	Lose rest of line

```
6.      1060    0     0     1          Select output 1
        121     89    0     5:4        Set style
        121     2     0     0          b2' = 0
        121     1     0     -19        Set count in B1
     1) 203     127   2     A2         If b2≠0→2) and b2' = b2-1
        121     2     0     4          b2' = 4 (second count)
        1101    90    0     A4/L1      Print NL
     2) 324     0     1     1:19       Next number to accumulator
        1101    90    0     A1/L1      Print next number
        201     127   1     A1         Count and repeat

7.   1) +1                             (A1) = 1
     2) +0                             Working space for partial
                                         sum
        1060    0     0     1          Select output 1
        1050    0     0     1          Select input 1
        121     1     0     1          Start count in B1
        121     90    0     2*         Set link in B90
     3) 121     127   0     A1/L100    Read next number
        320     0     0     A1         Add one to it
        234     127   0     A4         Jump if accumulator = 0
        320     0     0     A2         Add in partial sum
        321     0     0     A1         Subtract one
        346     0     0     A2         Store new partial sum
        201     127   1     A3         Count and repeat
     4) 121     81    1     -1         Compute n in B81
        121     88    0     3:0.2      Set style for integer
                                         print
        121     90    0     2*         Set link
        121     127   0     A2/L1      Print n
        324     0     0     A2         am' = (A2)
        121     89    0     6:3.0      Set style
        121     90    0     *2         Set link
        121     127   0     A1/L1      Print sum
        121     90    0     *2         Set link
        121     127   0     A4/L1      End line
        - - - - Next Instruction

8.      Omit line in store locations A1, 2A3, 3A3, 5A3.
        Replace "4)" by "20/L100)".

9.      * = 0
        A28/L1 = 1.2                   )Set parameters for 'comma'
        A29/L1 = 0.1                   ) and 'space'
        1050    0     0     6          Select input 6
        121     1     0     9          Set count in b1
     1) 1101    90    0     A1/L100    Read x to am
        121     89    0     1:4.3      Style
        346     0     0     A3         (A3)' = x
        1101    90    0     A1/L1      Print x
        237     127   0     A2         If x < 0 → 2
        1710    0     0     A3         am' = √x
        121     89    0     1:4.1      Style
        1101    90✓   0     A1/L1      Print √x
        203     127   1     A1         Count and repeat
        1117    0     0     0          End program
     2) 121     81    0     5.1        b81' a "I"
        121     90    0     -2A2       Set link
        121     127   0     A3/L1      Print I
     3) +0                             Working space
        E-2A1                          Enter directive
```

```
10.    R2
     1)374     0     0     A2          am' = θ/180
       362     0     0     A3          am' = θπ/180 = θ̄
       346     0     0     A4          (A4)' = θ̄
       1730    0     0     A4          am' = sin θ̄
       346     0     0     A99/R0      (A99/R0)' = sin θ̄
       1732    0     0     A4          am' = cos θ̄
       346     0     0     1A99/R0     (1A99/R0)' = cos θ̄
       121     127   50    0           Return to main program
     2)+180                            (A2) = 180
     3)+3.141593                       (A3) = π
     4)+0                              Working space

11.    * = 20
       A25/L1 = 3.5                    Set parameter for print-
                                         ing "+"

     1)-10                            )
    99)+0                             )Working space
       +0                             )and constants
     2)+10                            )

       1060    0     0     3           Select output 3
       121     1     0     18          bl' = 18
   →3)324      0     0     A1          am' = (A1)
       320     0     0     A2          am' = am + 10
       356     0     0     A1          (A1)' = am
       121     89    0     3:0.6       Set style
       1101    90    0     A1/L1       Print θ
       1101    50    0     A1/R2       Enter subroutine
       121     89    0     2:5         Set style
       324     0     0     A99         am' = sin θ
       1101    90    0     A1/L1       Print sin θ
       324     0     0     1A99        am' = cos θ
       1101    90    0     A1/L1       Print cos θ
    ---203     127   1     A3          Count and repeat
       1117    0     0     0           End program

       (Subroutine R2)

       E 1A2                           Enter directive

12.    121     81    0     J2K701.7
       1101    90    0     A3/L1

    or 121     81    0     J2K363.4
       1101    90    0     A3/L1

       010,000,000,000,111,000,001,111
    or 010,000,000,000,011,110,011,100

13.  7)C
       TAPE FAULT Ø ≠ 7
       A8 = -0.1*
       121     21    0     A7
       121     22    0     (A8-A7)U3+1
```

```
14.      Any time before label 3) write
         T3
         (Sp)θ(3 sp) sin θ(5 sp) cos θ
```

```
15.
 After 1050    0      0      6
  add  121     89     0      A4          b89' = A4
       1101    90     0      A6/L100     Read text to store A4
                                           onwards

 After 203     127    1      A1
  add  121     89     0      A4          b89' = A4
       1101    90     0      A6/L1       Write text
       121     87     0      3.5         b87' = 3.5
       1101    90     0      A5/L1       Write 13 N.L.'s

         Then label the enter directive thus:-

      4)E-2A1
```

```
16.      1050    0      0      5           Select input 5
     ┌-2)1053    127    0      A1          If binary →1)
     │←──1054    1      0      A2          Read to bl. Jump to 2)
     │                                       if c.c.c.
     │   113     1      0      A10         Store in A10
    ┌┼── 121     127    0      A3          Jump to 3)
    │└→1)1054    1      0      A2          Read to bl. Jump to 2)
    │                                        if c.c.c.
    │    113     1      0      A11         Store in A11
    │    1054    1      0      A3          Read to bl
    │    113     1      0      0.4A11      Store in .4A11
    └→3) - - - - Next Instruction

         N.B.   The address A3 in store line -2 A3 is meaningless
         a fault has occurred.
```

```
17.      1063    0      0      4           Delete output 4
         1060    0      0      4.1         Select output 4, binary
         1064    0      0      0.1         Output m.s. ½ of "M"
         1064    0      0      3.5         Output R.S. ½ of "M"
         1065    0      0      0           End record
         1071    0      0      4           Break output 4
```

```
18.a)    1050    0      0      3           Select input 3
         1060    0      0      5           Select output 5
         1057    5      0      A3          Read record to A3 onwards
         170     5      0      1.4        )Jump to 1) if more than
     ┌── 227     127    0      A1         )12 characters read
     │   1067    5      0      A3          Write complete record
    ┌┼── 121     127    0      A2          Jump to 2)
    │└→1)121     5      0      1.4J4       Reset b5
    │    1066    5      0      A3          Write twelve characters
    │                                        only
    └→2) - - - - Next Instruction
```

b) Before instruction 2) write

1065	0	0	2.5	Output 5 NL's.

13.6 Solutions to Magnetic Tape Operations.
Fixed Length Transfers (9.1).

1.	1001	2	0	1	Search
	1002	2	0	10:0.4	Read forwards

2.	1001	1	0	6	Search
	1003	1	0	1:0.2	Read backwards

3.	1001	1	0	1	Search for section 1
	1002	1	0	10:0.3	Read four blocks from tape
	121	1	0	.4-4:	Modifier-count in B1
	121	2	0	0	Modifier in B2
	1)101	3	1	14.-.4	Transfer half-
	113	3	2	1000	Words to store
	124	2	0	1	Advance modifier
	200	127	1	A1	Count and repeat
	1117	0	0	0	End program

4.	1001	1	0	4	Search
	1004	1	0	1:0.1	Write 1 & 2 to tape
	1005	1	0	0	Move tape forwards
	1004	1	0	3:0.0	Write 3 to tape
	1006	1	0	0:0.6	Move tape backwards
	1004	1	0	4:0.1	Write 4 & 5 to tape

5.	1001	1	0	5	Search
	1002	1	0	1:	Read one block from tape
	121	1	0	511	Modifier-count in B1
	121	2	0	0	Modifier in B2
	6)334	0	2	1:	)Reverse words
	356	0	1	2:	)of block
	124	2	0	1	Advance modifier
	203	127	1	A6	Count and repeat
	1001	1	0	5	Search
	1004	1	0	2:0.0	Write reversed block to tape

6.	121	121	2	0	Tape number to B121
	1001	122	1	0	Search
	1004	122	0	0.4 A1	)Write 10 blocks
	1004	122	0	5.4 A1	)to tape

7.
```
       1010    2      0     A10        Mount
       1002    2      0     A1         Read next block to Al
    20)1002    2      0     A2          "    "     "    "  A2
       121     1      0     -4D1       Modifier count in B1
       101     10     1     A10+4D1   )Compare titles
       102     10     1     A1+4D1    )
       215     127    10    3*         Jump if titles disagree
       200     127    1     -3*        Jump to continue comparison
       121     127    0     A21        Jump if titles agree
       1002    2      0     A1         Read next block to A1
       121     1      0     -4D1       Reset B1
       101     10     1     A10+4D1   )Compare titles
       102     10     1     A2+4D1    )
       215     127    10    A20        Jump if titles disagree
       200     127    1     -3*        Jump to continue compar-
                                         ison
    21)1006    2      0     0          Move tape back 2 sections
       121     127    0     3 A10      Exit
    10)C
       MERGER INC. TAPE 2AB
```

8.
```
       JOB
       MEAN VALUE CALCULATION
       TAPE
       1 XYZ - 3
       TAPE
       2 XYZ - 4
       COMPILER ABL
       * = 0
       1001    1      0     1          Search
       1001    2      0     1          Search
       1002    1      0     3:0.3      Read 4 blocks to 3:
       121     2      0     -3         Modifier-count in B2
     3)121     1      0     511           "      "    in B1
       346     0      0     J4         Clear accumulator
     2)320     0      1     3:         Add in next number
       203     127    1     A2         Count and repeat
       1574    0      0     512        Take average
       346     0      2     7:3        Store average
       201     127    2     A3         Count and repeat
       1004    2      0     3:0.4      Write 5 blocks to tape
       1117    0      0     0
       E0
       ***Z
```

9.
```
       1024    1      0     A1         Read P1
       1024    2      0     A2         Read P2
       101     1      0     A1        )
       104     1      0     A2        )Compute |(P1+P2)/2|
       163     1      0     0         )
       1001    1      1     0         )Search to |(P1+P2)/2|
       1001    2      1     0         )
       121     4      0     -49        Count in B4
    20)1003    1      0     0.4 A10    Read 1st set of 5 blocks
       1003    1      0     0.4 A11    Read next five blocks
                                              Cont.........
```

9. cont.

121	3	0	4:	)
1004	2	3	A10	)Write 5 blocks in
122	3	0	1:	)reversed order
216	127	3	-2*	)
214	127	4	2*	Avoid unnecessary read
1003	1	0	0.4 A10	Read next 5 blocks
121	3	0	4:	)
1004	2	3	A11	)Write 5 blocks in
122	3	0	1:	)reversed order
216	127	3	-2*	)
201	127	4	A20	Count and repeat

Variable Length Transfers (9.2).

1. 1)H+7,+173

1044	6	0	A1	

2.

1001	2	0	1	Search
1032	2	0	6:0.1	Write buffer
121	4	0	750.1	b4' = 750.1
1040	4	0	A3	Transfer 750 words

3.

1001	2	0	1	
1032	2	0	6:01	
1042	0	0	0.3	"3" Marker
121	4	0	750.4	"4" Marker
1040	4	0	A3	Transfer 750 words

4.a) 5)H+4,+6

	1044	3	0	A5	Search
b)	1030	3	0	A6B+1:	Read buffer
c)	121	1	0	100	b1' = 100
	1040	1	0	A1	Transfer 100 words
d)	121	2	0	0.3	b2' = .3
	1040	2	0	A2	Transfer till marker ≥ 3
e)	127	2	0	K.7'	b2' = Integral pt. b2
	1040	2	0	A3	Transfer b2 words
f)	121	3	0	200.4	b3' = 200.4
	1040	3	0	A4	Transfer 200 words

5. 1)H 100, 3

1044	4	0	A1	Search
1030	4	0	(*+1:)B	Read buffer
1032	4	0	(*+1:)B	Write buffer
1042	4	0	0.3	Mark "3"
1043	4	0	0	Release buffer

```
6.a)   1001   3     0    1          Search
       1030   3     0    (*+1:)B+0.1 Read buffer
       121    1     0    0          b1' = 0
       1040   1     0    A2         Transfer first string
       121    2     0    0          b2' = 0
       1041   2     0    0          Skip second string
       121    2     0    0          b2' = 0
       1041   2     0    0          Skip third string
       121    2     0    0          b2' = 0
       127    1     0    K.7'       b1' = integral pt. b1
       1040   2     1    A2         Transfer fourth string

  b)   1043   3     0    0          Release buffer


7.     1031   6     0    2:         Read backwards buffer
       121    5     0    100.7      b5' = 100.7
       1040   5     0    216        Transfer 100 words


8.     A1 = (*+1:)B                 Set A1 for next block
       1001   1     0    1          Search
       1034   1     0    A1+0.1     Buffer
       121    1     0    700        b1' = 700
       1040   1     0    A1+2:      Read 700 words


9.a) 3)+0
     4)H+100,+4
       1001   1     0    1          Search
       1030   1     0    6:         Buffer
       121    1     0    0          b1' = 0
       1040   1     0    A2         Read
       1024   1     0    A3         (A3) = current psn.

  b)   1044   1     0    A4         Search for (A4)
       1032   1     0    7:         Buffer
       121    2     0    160.3      b2' = 160.3
       1040   2     0    A7         Write 160 words

  c)   1044   1     0    A3         Search for (A3)
       1030   1     0    6:         as a)
       121    3     0    0          as a )
       127    1     0    K.7'       b1' = integral pt. b1
       1040   3     1    A2         Read string


10.    R1
     1)1036   121   0    0          b121' = selected tape = n
       143    121   0    0          b121' = n/2
       121    80    0    0          b80' = 0
       1033   4     0    0          Select tape 4
       1040   80    0    A12        Read to A12
       1033   122   0    0          Select tape 4
       121    127   90   0          Return to main program
       Z

Exit from main program will be
       1101   90    0    A1/1
```

11. 1)+0

```
    1037   1     0    A1          (A) = mode of tape 1
    121    10    0    3           b10' = 3
    102    10    0    A1          b10' = 3- (A1)
 ┌─ 214    127   10   *4          If b10 = 0, jump
 │  1032   1     0    *B+1:       Buffer
 │  121    1     0    30          b1' = 30
 │  1040   1     0    A7          Write 30 words
 └→ - - - - Next Instruction
```

12.

```
    1001   6     0    1           Search
    1030   6     0    (*+1:)B     Buffer
    121    1     0    0           b' = 0
    1041   1     0    0           Skip to first marker,
                                    b1' = count
    127    1     0    K.7'        b1' = integral pt. b1
    143    1     0    0           b1' = b1/2
    1001   6     0    1           Search
    1030   6     0    (*+1:)B     Buffer
    1040   1     0    A1          Read b1 words to A2
```

13.7 Solutions to Job Document exercises (10)

1.

	Job Description	Program	Data
a)	JOB MEAN CALCULATION INPUT 1 X3 RESULTS STORE 6 COMPILER ABL (Program) ***Z	Included in Job description	DATA X3 RESULTS (Data) ***Z
b)	JOB MONTE CARLO RALLY 1962 INPUT 0 RALLY ANALYSIS 1 RESULTS 1962 2 WEATHER CONDITIONS OUTPUT 1 CARD 2 LINE PRINTER 4 BLOCKS DECKS 4 TAPE 1 ENGINE SPECIFICATIONS 1962 TAPE FREE 2 RALLY RESULTS 1962-1967 TAPE/1 3 RALLY RESULTS 1950-1955 TAPE/2 END 3 RALLY RESULTS 1956-1961	COMPILER EMA RALLY ANALYSIS (Program) ***Z	DATA RESULTS 1962 (Data) ***Z and DATA WEATHER CONDITIONS (Data) ***Z

	Job Description	Program	Data
1.b) cont.	COMPUTING 10 MINUTES EXECUTION 15 MINUTES STORE 25 ***Z		
c)	JOB F/126/JKB 8 Feb.63 INPUT 0 F/126/JKB 1 SPECIAL RANDOM NUMBERS SELF = 2 OUTPUT 1 TAPE FREE/FIVE-HOLE TELETYPE 20 BLOCKS 126 CLASSIFIED RESULTS COMPUTING 25 SECONDS STORE 3 DATA (Data) ***Z	COMPILER ABL F/126/JKB (Program) ***Z	DATA SPECIAL RANDOM NUMBERS (Data) ***Z
d)	JOB SURVEY ANALYSIS 14 INPUT 0 SURVEY ANALYSIS 1 SURVEY RESULTS OUTPUT 1 TELETYPE 2 BLOCKS COMPUTING 20 SECONDS ***Z	COMPILER ABL SURVEY ANALYSIS (Program) ***Z	DATA/1 SURVEY RESULTS (Data) ***Z and DATA/2 END SURVEY RESULTS (Data) ***Z
e)	JOB SURVEY ANALYSIS 15 INPUT TAPE S5/3/500 0 SURVEY ANALYSIS 1 SURVEY RESULTS/2 OUTPUT 1 TELETYPE 2 BLOCKS COMPUTING 20 SECONDS ***Z	None Needed	DATA SURVEY RESULTS/2 (Data) ***Z

2.a) Input data tapes with headings:-

COPY TAPE FREE
DATA FOR MONTE '62

DATA
RESULTS 1962
(Data)
***Z

	Job Description	Program	Data

2.a) cont

and

COPY TAPE 101
DATA FOR MONTE '62

DATA
WEATHER CONDITIONS
(Data)
***Z

Ensure that they are input in this order since the first gives a title to a free tape and the second has to be stored from block 101 of the same tape.

b) JOB
MONTE CARLO RALLY 1962

INPUT
0 RALLY ANALYSIS
TAPE 6/1/0
1 RESULTS 1962
TAPE 6/101/0
2 WEATHER CONDITIONS

DECKS 5
TAPE
6 DATA FOR MONTE '62
(Output and other tapes as before)

***Z

The increase in "decks" assumes that the input tape will be in use at the same time as the other tapes at some point in the program.

c) PRINT TAPE
126-CLASSIFIED RESULTS

d)1) COPY TAPE FREE
DATA FOR MONTE '62

DATA
RESULTS 1962
(Data)
***C
DATA
WEATHER CONDITIONS
(Data)
***Z

2) 'INPUT' section would read:-

INPUT
0 RALLY ANALYSIS
TAPE 6/1/0
1 RESULTS 1962
TAPE 6/61/413
2 WEATHER CONDITIONS

	Job Description			Program	Data
3.	JOB				
	F1/JKB/MEAN CALCULATION				
	INPUT				
	1 ASSORTED NUMBERS				
	STORE 1				
	COMPILER ABL				
1)	+0				
	+10				
	1050	0	0	1	Select input 1
	121	1	0	9	b1' = 9
→2)	1101	90	0	A1/L100	Read to am
	320	0	0	A1	am' = am + (A1)
	346	0	0	A1	(A1)' = am
	203	127	1	A2	If b1 ≠ 0 subtract one & → 2)
	324	0	0	A1	am' = (A1)
	374	0	0	1A1	am' = am/10
	121	89	0	4:2.2	Set Style
	1101	90	0	A1/L1	Print
	1117	0	0	0	End
	E2A1				Enter
	***C				End of document

DATA
ASSORTED NUMBERS
(Ten numbers to follow)
***Z

This uses only one tape for all documents. If the data tape is separate from the rest *** C must be replaces by *** Z.

13.8 Solutions to Advanced Programming Exercises

a) Store Extracode Operations (11.1).

1.	1160	0	0	A1	Read block A1
	121	1	0	A2	b1' = A2
	1163	1	0	A1	Duplicate A1 as A2
	121	2	0	A3	b2' = A3
	1164	2	0	A1	Rename A1 as A3
	1164	0	0	A3	Release A3 to the drum
	1167	0	0	A2	Lose block A2
2.	121	1	0	A3	b1' = A3
	1164	1	0	A1	Rename A1 as A3

```
3.      A1 = 2:
        A2 = A1 + (P1 × 512) - 1

    10)+0

        121     1       0       0           Clear Modifier
        1160    0       0       A1          Read A1 to core store

 ┌────→ 172     1       0       -1P1×1:    )If last block has been
 │ ┌─── 224     127     0       2*         )read → A21
 │ │    1160    0       1       A1+1:       Read next block
 │ └──→ 121     2       0       -255        Set count to -255

 │┌→20)324     0       1       A1         )
 ││     362     0       1       A1+1       )Add x_i x_i+1
 ││     320     0       0       A10        )into A10
 ││     356     0       0       A10        )

 ││     124     1       0       2           Increase modifier by 2
 │└──── 201     127     2       A20         Repeat 256 times

 │      1167    0       1       A1-1:       Lose used block
 │      172     1       0       P1×1:      )Repeat P1 times
 └───── 225     127     0       -4A20      )
        - - - - Next Instruction

4.      1160    0       0       A10         Read first block to core
        121     1       0       -100:       Set modifier / count
        121     2       0       A1         )Set b2 and b4
        121     4       0       A2         )for renaming

  ┌→20)124     1       0       1:          b1' = b1+1:
  │ ┌── 214     127     1       2*          Jump if last pass
  │ │   1160    0       1       A10+100:    Read next block
  │ └─→ 1164    2       1       A10+99:     Rename current block as
  │                                           A1

  │             < Process A1 into A2 >

  │     121     3       1       A11+99:     Set b3 for renaming
  │     1163    3       0       A2          Rename A2 as A11+99:+b1
  │     1164    4       0       A1          Rename A1 as A2(*)
  │     1161    0       1       A11+99:     Release A11+99:+b1 to
  │                                           drum
  └──── 215     127     1       A20         Repeat 100 times

        1167    0       0       A2          Lose A2

        (*) This instruction avoids a non-equivalence interrupt,
        with possible attendant hold up, in the main processing.

5.      121     4       0       -99         Block counter
        121     10      0       A1          Set for renaming
        121     1       0       0           a = o
        1160    0       0       10:         Read block 10: from drum
                                                        Cont.........
```

```
→100)1164   10    1    10:      Rename block 10:+a as A1
     214    127   4    2*       Jump for last pass
     1160   0     1    11:      Read next block (11:+a)

     121    2     0    -511     )
  ┌→ 1730   0     2    511A1    )Replace α by sin α
  └─ 201    127   2    -1*      )in A1

     1161   0     0    A1       Release A1 to drum
     121    3     1    10:      b3 = 10:+a
     1164   3     0    A1       Rename A1 as x0:+a
     124    1     0    1:       a = a+1:
 ─── 201    127   4    A100     Repeat 100 times
     121    127   40   0        Exit from routine.
```

b) Branching example (11.2).

Branch 0.

```
   1012    1      0      0.1A6    Mount tape one on channel
                                    one
   1012    2      0      0.2A7    Mount tape two on channel
                                    two

   1103    3      0      91D1     Permit 3 branches

   121     1      0      0        Set b1 as 'switch' for
                                    last block
   121     2      0      2:      )Set b2 and b4 for
   121     3      0      4:      )renaming blocks
   121     4      0      3        b3' = code number of
                                    store

   1104    2      0      A1       Establish branch 2 at
                                    A1


4) 1106    2      0      0        Wait till branch 2 is
                                    dead
   215     127    1      A5       If last block processed
                                    jump to A5

   1164    2      0      1:       Rename 1: as 2:
   1104    2      0      A1       Establish branch 2 at
                                    A1

   1101    90     0      A/5      Enter subroutine

   1106    1      0      0        Wait till branch 1 is
                                    dead
   1164    4      0      3:       Rename 3: as 4:
   1104    1      0      A3       Establish branch 1 at
                                    A3
   121     127    0      A4       Go to A4

5) 1106    1      0      0        Wait till branch 1 is
                                    dead
   1117    0      0      0        End program


   R5                            )
   -----                         )
   -----                         )Routine R5
   -----                         )
   121     127    90     0       )
   Z


   6)C
     STORE 3: SALES RECORDS
     SKO
   7)C
     AMENDMENTS TO SALES RECORDS/MARCH
     SKO
```

Branch 1.

```
→3)1002     1      0      5:        Read one block from tape
                                      1 to 5:
   101      5      0      5:0.4     b5' = code word of
                                      article
   150      5      0      4:0.4    )jump to A3 if code
   225      127    0      A3       )words disagree
   1006     1      0      0         Move back one section
   1004     1      0      4         Write block 4: to tape
   1105     64     0      0         Kill current branch  →
```

Branch 2.

```
→1)1002     2      0      1:        Read one block from
                                      tape 2 to 1:
   150      4      0      1:        bt' = (1:) - store
                                      code number
   224      127    0      A2        If bt = 0 jump to A2

   150      0      0      1:        bt' = (1:)
   225      127    0      A1        If bt ≠ 0, jump to A1

   121      1      0      1         Set switch to last
                                     block

 2)1105     64     0      0         Kill current branch  →
```

c) Monitoring and Trapping (11.3).

```
1.    1132     0      0      A4        set trap

   4)+0  +0  +0  +0  +0  +0  +0  +0  +0
     H  A3  0
     +0  +0
     H  A5  40D1
   5)1114   0      40     0
   3)
```

```
2.    1132     0      0      A4
      1112     0      0      A3        set private monitor
   4)+0  +0  +0  +0  +0  +0  +0  +0  +0  +0  +0  +0  +0
     H  A2  10D1
   2)1114   0      10     0.1
   3)121    3      0      -2
     121    89     0      1:9.3
  →5)334    0      3      3A1
     1101   90     0      A1/L1
     201    127    3      A5
     1117   0      0      0
```

```
3.    2)+2
        1123    0      0      1000       set local instruction
                                           count
        346     0      0      J4         clear acculuator
        324     0      0      A2
        346     0      0      99
 ┌→  1)346     0      0      J4
 │      324     0      0      99
 │      1520    0      0      1          am' = am+1
 │      346     0      0      99
 │      1730    0      0      99         am' = sin(99)
 │      346     0      0      100
 │      1714    0      0      99         am' = '/(99)
 │      320     0      0      100
 │      1521    0      0      1          am' = am-1
 └──── 1237    127    0      A1         jump if am ≤ 0
        346     0      0      J4
        324     0      0      99
        121     89     0      6:0.2     )print the contents
        1101    90     0      A1/L1     )of the accumulator
        1117    0      0      0

4.      1132    0      0      A3
        121     3      0      0
 ┌→1)1004      4      3      40:0.0
 │      124     3      0      1:
 │      172     3      0      20:
 └──── 225     127    0      A1
        1117    0      0      0
     2)121     81     3      0
        1011    5      0      A10
 ┌→4)1004      5      3      40:0.0
 │      124     3      0      1:
 │      172     3      0      20:
 └──── 225     127    0      A4
        1344    81     0      9          logical shift right of
        120     81     0      20         b81 by 9 places.
        121     88     0      2:0.2
        1101    90     0      A2/L1
        1117    0      0      0
   10)C
        OVERFLOW BLOCKS
        SK00
     3)+0   +0   +0   +0   +0   +0   +0   +0   +0   +0
        H   A2   0
        +0   +0   +0
```