ICT 1301
MPL_2
MANUAL
Scanned Feb 2010
by the
1301 Resurrection
Project
ict1301.co.uk

Chapter 5

INPUT/OUTPUT CONCEPTS

TAPE

The tape package must be requested on the header card when tape functions are used or when the program sector facility is used.

PERIPHERALS (Card Reader, Card Punch and Printer)

Two fundamentally different input/output systems may be employed by the assembled program. These are entirely interchangeable, and have been written so that both have the same set of entry points.

These two systems are incorporated in the standard control packs A and B respectively, and comprise:

System A — A print, punch and feed control

System B — A sequential system whereby either cards are read or punched, or lines printed, in batches, thereby minimizing waiting times.

System A is likely to give faster running if the job uses the input/output units at random, the second if the job uses any one peripheral unit in long bursts.

Using either system, the source program Read/Print/Punch instruction causes nothing but an entry to a control program which may or may not promote an input and/or an output. Thus, if an error stop occurs while processing a batch of cards, it is not possible to calculate precisely which individual card in the batch was responsible for the error stop. This need cause no difficulty, but it is advisable to arrange that error stops display in the registers information that indicates the data currently being processed. Also, this implies that at the end of the job, the contents of the output buffers must be emptied. For this purpose, the Stop instruction can additionally cause a jump to the runout entry of the input/output control. In both cases the buffers are on the drum.

System A

The P.P.F. system is based on 1300 series input/output speeds (in the ratio 6:6:1, but will work with the ratio 3:3:1). Physical input/output occurs when either:

(a) A card read order is given with no cards in the input buffer or,

(b) An output order causes the appropriate buffer to become completely full.

At such a point, depending on the state of the buffers, the computer will read up to 7 cards, print up to 6 lines and punch one card.

System B

The sequential system will process 10 cards or print 10 lines in any one burst. Physical reading takes place when a read order is given with the read buffer empty. Physical punching or printing occurs when an order fills the last space in the punch or print buffer.

On a runout entry, the punch and print buffers are emptied. If any printing has occurred during the run, approximately 30 inches of paper are thrown. Similarly, if any punching has occurred, a blank card is punched to check the last card.

PAPER SUPPLY TEST

A program indicator is provided on 1300 series machines to test if the paper trolley is empty. It need not be tested for every line, or indeed at all, but if the test is required, digit 1 of the space requirement (WS20) should be set to 9, so that when the trolley is empty, a distinctive stop is met. On restart, the appropriate spacing will still be performed before printing; thus if the print is paged, the new paper should be set up in the same position on the sprockets as the forms just removed by the operator.

The operator should be warned if this test is not made, so that attention can be given to the paper supply and the exhaustion in supply immediately noticed.

WHICH SYSTEM TO USE

As mentioned above, if the accesses to units are random, the P.P.F. system will probably be fastest in use, because of its time-sharing. It is somewhat slower in control time than the sequential system however and reads cards in batches of only 6 or 7 compared with the batches of 10 of the sequential system. Thus if spells of reading followed by spells of printing or punching occur, the sequential system is likely to be faster. As the systems are fully compatible, both could be tried in succession in doubtful cases.

3-BANK PRINTING

Although a standard 1300 employs two-bank printing, both of these input/output systems use three-bank print programs. This is necessary because the distribution generator distributes into 120 positions and cannot be reset for 80 positions. The only loss of time is 1.3 milliseconds per line due to clearing extra print area, and a three-decade transfer instead of a two-decade transfer to the buffers (1 minute in 45,000 lines).

CHARACTER SETS

For reading and punching, the standard I.C.T. 5-zone code is employed; while for printing, the standard machine character set is used.

Chapter 6

PROGRAMMING TECHNIQUES

INTER - BLOCK TRANSFERS

There is no automatic means provided by the Assembler for transferring program control from the end of one block of program to the next block of program. The last MPL 2 instruction in each block must therefore be a 'J' instruction, and should be either a jump back into the present block of program or a jump to a new block of program. The only exception to this rule is when the end of the block of program is also the final instruction to be obeyed in the complete program, in which case the last instruction in block can be '⊡' with the runout facility used. Any failure to comply with this rule will mean that program control will pass into the constants in that program block and such effect will be unpredictable.

INITIAL TESTING

Every program block is limited to 200 words. In the initial testing stage it is quite possible that a block (as specified) occupies more than 200 words. This limits the usefulness of a test considerably and wastes the machine time used in such an assembly. In the early stages therefore, block sizes should be kept fairly small in order to see how many instructions are generated by the Assembler. When the size of the program is known and the testing has reached a fairly advanced stage, then optimum block sizes can be fixed.

The instructions which are most likely to occur in overfull blocks are the three input/output functions. Estimating the coding produced by these functions is very difficult and extra care should be taken with program blocks where they are used. It is suggested that, as part of initial testing, the input/output statements should be assembled independently, one per block, merely to ascertain the size of program produced.

SEGMENTING THE PROGRAM

The division of the program into a number of program blocks is a very important part of assembling a program. If this is badly performed then the time taken to run the object program will be increased.

An example of the simplest case is that all related program should be together. If there is a major path through a program (i.e. that performed by the majority of cases), then it is this path that should form the continuous string of program instructions and it should be the exceptions that appear in other program blocks. Further, when there is a loop in the program (a section of the program performed several times on each program pass), the loop should be confined to one block of program if possible. The following example will make it more evident.

A program reads 125 cards and prints 125 lines a minute. It contains a program loop which is performed three times for every card read. This loop stretches across two program blocks but by manipulation of program the loop can be contained in one block.

Immediately six changes of program block have been saved and, as the time taken for one transfer of program from the drum is 12 milliseconds:

time saved is 72 milliseconds for every card read and for every line printed

time taken for 1 card and line @ 125 cards and lines a minute = 480 milliseconds

time now taken = 480 - 72 = 408 milliseconds.

This is approximately 150 cards and lines a minute and would give a 20% increase in the speed with which the program runs thus saving 16.7% on the computer time to run the program. Time can also be saved, where a large enough I.A.S. configuration exists, by utilizing the facility for storing more than one program block in I.A.S. at one time. As fewer drum-to-I.A.S. transfers are required, considerable time will be saved where inter-block jumps are involved.

MODIFICATION OF J INSTRUCTIONS

Normally there will be no need to modify a 'J' instruction. If it is necessary to use this facility, care must be taken to understand the exact way in which it works. It is *not* the label number which is modified by a Modify instruction, but the I.A.S. address which will be the operand of the generated 1300 instruction. In the following example assume that the value of COUNT is 1 and the I.A.S. address of label 12 is 30.

F	OPERAND
¼	C O U N T
J	0 0 - 1 2

The effect of this pair of instructions is that program control passes to address 031 not to label 13. This technique can be used to enter subroutines where there are alternative entry points. This technique can be used only when the object label is in the same program block as the 'J' instruction.

SUBROUTINES

Machine-code programs are included in an MPL 2 program by using the 'E' function. Since the Assembler will perform the sequence check on any machine-code program cards, these program cards must be sequence-numbered to agree with the preceding and following MPL 2 cards.

If a subroutine is to be included in a block of program then a jump must be made to a label which precedes that subroutine and the subroutine must begin with a '41' instruction.

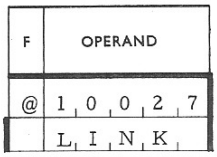
F	OPERAND
J	0 0 - 3 5
L	- - - 3 5
E	X 2 4 1 0
	etc

This method will apply only when all references to a subroutine are in the same block as the complete subroutine.

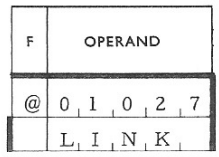
GLOBAL SUBROUTINES

It is sometimes necessary to use a subroutine in a program block other than the block in which that subroutine originally appeared. This type of subroutine is called a global subroutine.

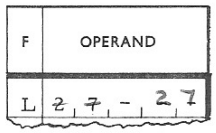
The new function explained in Chapter 4 (see '@' function) simplifies the inclusion of such subroutines. If there is a subroutine under label 27 in program block 3 which is performed by both program blocks 1 and 2, then the diagram below will illustrate the use of this function.

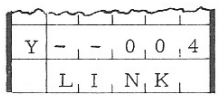


Program Block 1



Program Block 2



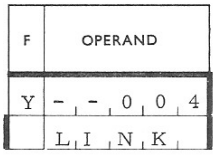


Program Block 3

It should be noted that:

(a) the subroutine does not commence with a '41' instruction,

(b) the last instruction of the subroutine to be obeyed (or if the program is written in machine code, the first MPL 2 instruction following) must be a 'Y' function of the form:



where LINK is the data name in which the return mechanism has been placed by both of the '@' functions.

The return transfer will be to the MPL 2 instruction following the function which caused entry to the subroutine..

SUBROUTINE IN DATA STORE

The placing of subroutines in I.A.S. normally used for data storage can only be achieved with a knowledge of the mechanics of assembling. The one, constant, relevant factor is that the allocation of data storage always commences at I.A.S. location 360 in tape programs and 115 in non-tape programs.

If a programmer has a routine that is in constant use throughout the program then storage would be wasted if this routine appeared in every block of program in which it is used. The alternative therefore is to keep this routine in the data storage area where it will be permanently available. This requires two essential actions on the part of the programmer. At the beginning of the very first block of program, an area of storage equal to the length of the routine must be allocated a fictitious name. This will ensure that the area allocated to the subroutine commences at 360 or 115. Immediately after label 0 (the entry point of the program) the 'V' function must be used to set values to the data name that has been used. The values will be 1300 instructions and the operands must have absolute addresses.

F	OPERAND
%	- , - , - , - , 1
6	S , U , B , - , R

L	- , - , - , - , 0
V	- , - , - , - , 6
	S , U , B , - , R
4	1 , 0 , 3 , 0 , 0
6	7 , 0 , 2 , 9 , 9
	, , etc.

In order to enter the subroutine stored at 360 or 115, which has been called S U B - R, the 'Y' function is used.

F	OPERAND
Y	- , - , 0 , 0 , 4
-	S , U , B , - , R

A number of routines can be set in data storage if necessary. Each one must have its own name which is defined at the beginning of the first block and each one must have a value set to it immediately following label 0. This method of handling subroutines is restricted by the amount of data used in the calculation and housekeeping and will probably only apply to very small routines. Care must be taken in using this method that the addresses used on such routines are the correct absolute addresses.

READ FUNCTION

There are two kinds of Read functions, which have been described earlier, and they are used in the following way. The Read, non-distribute (which will be called control), can be made anywhere. The amount ~~is~~ of coding produced is very small. It generates a jump to the card routine, a test for the end of file marker ($\overline{13}$, in columns 3 to 8), and isolates the designation field as specified by the programmer and leaves the designation field in register B. The programmer can then investigate the value of the designation field and jump to the appropriate distribution. The distribution for a particular card type can be followed by the calculation and housekeeping performed on that card. In other words, all parts of the program related to a particular card type are very close together and should be in the same block.

PRINT FUNCTION

The Print function has two instructions in order to separate the distribution and physical print. There are two reasons for this; firstly, there may be some heading lines which are used frequently in a program, in which case their distributed form for output has a fixed value and the input/output area can be set equal to this value using the 'V' function. This saves any time that would be spent in distributing the line of print during running time. The second reason for the separation of the distribution and physical print is to allow the programmer to insert any editing symbols into the line of print.

These comments also apply to the Punch function except that it is not usually necessary to edit punched output.

Chapter 7

ASSEMBLING AND TESTING PROGRAMS

THE MPL 2 PROGRAM PACK

The MPL 2 program pack is produced by punching the information written on the program sheets (see Chapter 4) into the individual program cards. Prior to the program being assembled however, it is essential to ensure that the following conditions are observed:

(a) 'This card sequence number' for the first card to be presented to the Assembler must have a value of zero.

(b) The Label instruction beginning the first ~~sector~~ block OF ANY SECTOR ~~presented to the Assembler~~ must not have a value in its second character.

(c) The final MPL 2 instruction of the pack must be 'L B E - -'.

(d) The Job Set-up cards must be ready.

ASSEMBLING AN MPL 2 OBJECT PROGRAM

An MPL 2 source program is assembled by the assembly program which is read on to the drum from the MPL 2 master tape via Job Set-up. The MPL 2 source program pack is placed in the card reader immediately behind the Job Set-up cards.

The assembly program is entered, the MPL 2 program pack read, and the resultant object program is stored on tape. During this assembly it is possible to obtain a printout of the source program listing the addresses of data names and labels. Similarly, a request may be made to print out the assembled object program. These requests are made by manual indicators.

If the program utilizes the sector facility, the optional printout of data names, their sizes and addresses, is at the end of each sector. The drum addresses of overfull blocks, and labels referenced but not used, are printed out at the end of each sector. The printout of tape error statistics during assembly is prior to stop 111329.

When the object program tape is ready to run, it is entered via Job Set-up, the first program sector to be transferred to the drum and to be subsequently obeyed being specified on the Job Set-up cards.

MANUAL INDICATORS USED DURING ASSEMBLY – *When SET*

Manual indicator 21	suppresses source program list.
Manual indicator 22	suppresses object program printout.
Manual indicator 23	indicates that the source program uses no tape and that the data name area now starts at 115.
Manual indicator 24	causes a single-sector object program to be punched out in fast-read format.
Manual indicator 25	prints out the data name table.
Manual indicator 26	inhibits tape error printing.

SECTORS AND DATA NAMES

When a program is too large to be accommodated on the drum, it may be split into convenient sectors by the insertion of end-of-sector labels at suitable points. Any sector can then be transferred to the drum by instructions contained in the Job Set-up cards. An MPL 2 object program can comprise up to 99 such sectors.

Where identical data names are used in sectors being successively processed, packages (tape and input/output) need only be specified in the header card for the first sector. In this case the I.A.S. is preserved during an inter-sector jump. (See Multiple Sector Program Running below.)

TESTING AN MPL 2 PROGRAM

Testing an MPL 2 program should proceed at a faster rate than for ordinary machine-code programs because there is less chance of minor coding mistakes, and the program storage is automatically handled.

Care must be exercised when checking the data names since a mispunched name will be treated as a new name and will be allocated a new word.

The assembly program checks for errors in the source language cards. When an error is found, the error type is printed out alongside the box in error.

STOPS LIABLE TO OCCUR DURING THE ASSEMBLY PROGRAM

CR3	Meaning	Action
111321	Card sequence error	Correct sequence start
111323	Too many data names	Abandon
111322	Misfeed	Refeed rejected card
111341	Overwriting tape or input/output package with object program	Start
111342	Object program too large for this reel	Abandon
111328	Label 0 missing	Start
111329	End of assembly	
111340	INVALID HEADER CARD	ABANDON
111345	NO INPUT/OUTPUT PACKAGE SPECIFIED FOR NON-FIRST SECTOR	START

STOPS LIABLE TO OCCUR DURING THE OBJECT PROGRAM RUNNING

111353	Record too big for file	Abandon
111350	No tape allocated for inter-sector jump	Fit tape, start or abandon
111351	Sector not on program tape	Abandon
111352	Discrepancy of program block counts	Re-run job

If an unexpected stop should occur and the programmer wishes to runout his print and punch buffers, the following should be encoded in the control register and be obeyed.

F	ADDRESS	
00	4, 0, 1, 7	CR 1
11	1, 3, 5, 9	CR 2

3139(8.64)

RUNNING THE OBJECT PROGRAM

When the program tape is ready to run, the first sector is loaded on to the drum via Job Set-up and then the program is entered. Any data cards are placed after the Job Set-up cards.

Note When object programs are being ~~run~~ LOADED, indicator 20 or 21 must be set when ~~the Job Set-up 'E' card is read.~~ CARDS ARE READ BEFORE THE OBJECT PROGRAM IS LOADED TO THE DRUM (ie. J.S.U E CARD OR THE OBJECT PROGRAM PACK)

PUNCH OUT OF OBJECT PROGRAM

This facility should only be used for proven object programs, as any errors found after punching will necessitate alteration to the source language cards, re-assembly of the corrected program, and hence the punching out of a new pack.

If the proven object program is to be punched out in the form of a fast-read pack, the operation is performed immediately following the assembly of the object program, i.e. as soon as the last block of the program has been written to the object program tape. It is not possible to punch out a program that is more than one sector long. With the object program, the packages required are also punched out.

RUNNING THE OBJECT PROGRAM WITH A FAST-READ PACK

When running an object program, Initial Orders are used to read in the fast-read pack plus any drum data. The pack is entered by an 'E' designation in the normal way. This assembly produced program pack is complete and it is not necessary to add any standard package to the punched program.

MULTIPLE SECTOR PROGRAM RUNNING

An MPL 2 program must contain either an input/output package or a tape package. All sectors containing one of these can be entered via Job Set-up or by an inter-sector jump. The sector will be entered at label 0 and all data and working stores will be zeroized on entry. Data areas will be allocated according to definitions within that sector.

It is possible however to assemble sub-sectors which have neither an input/output package nor a tape package. Entry can only be made to a sub-sector by an inter-sector jump and the state of all data and working stores will not be altered by such an inter-sector jump. Similarly the data allocations will be the same in the sub-sector as in the main sector which was assembled immediately before it.

Entries to main sectors will assume that the input/output buffers are empty. Hence, if such entries are made by inter-sector jumps, there must be a runout before the jump.

If inter-sector jumps are being made, the object program tape must be called PROGRAM-TAPE in its identity (as for Job Set-up) and this name should be unique among the tapes used for any job.

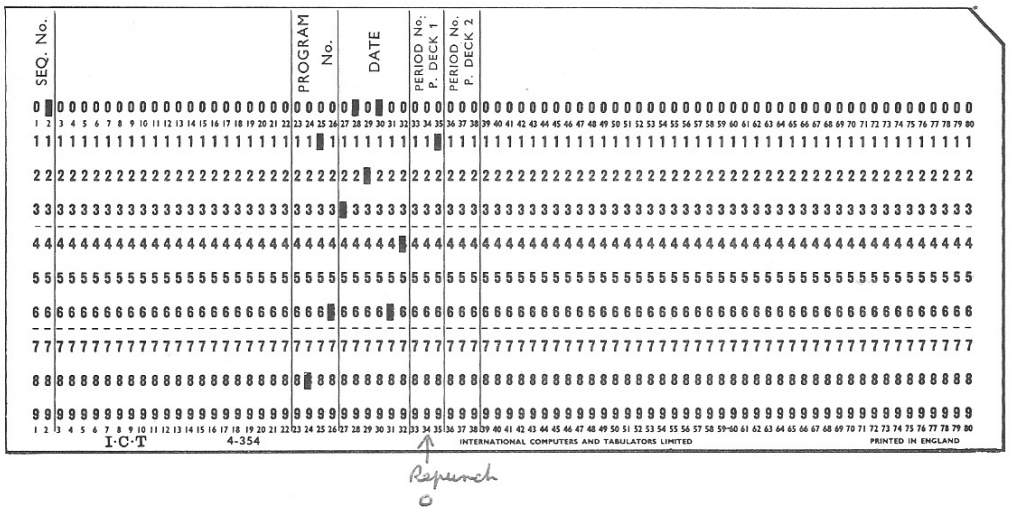
JOB SET-UP FOR MPL 2

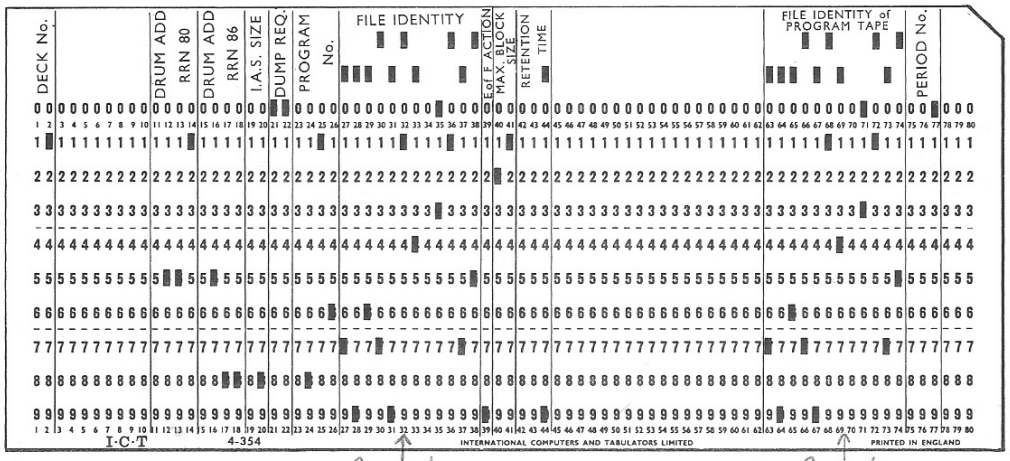
It is suggested that the appropriate chapter in the Tape Housekeeping manual is read.

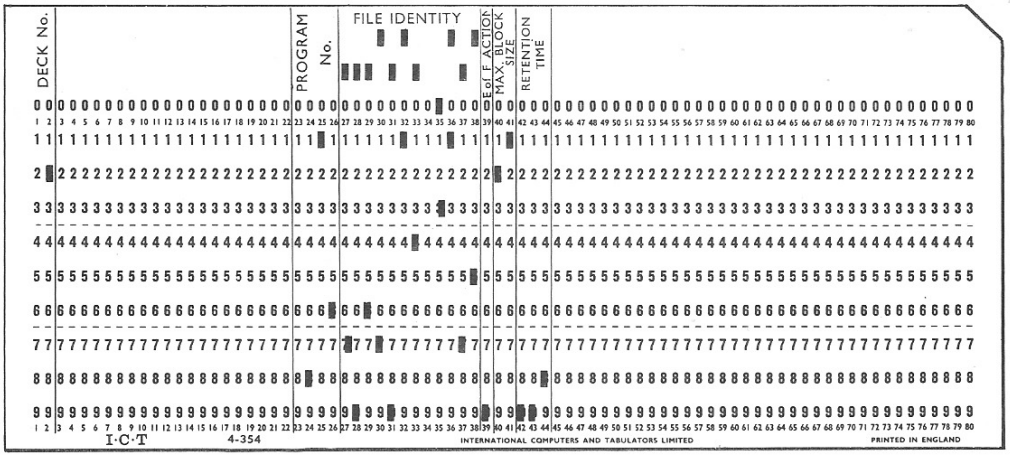
The following information is required for assembling and running the object program.

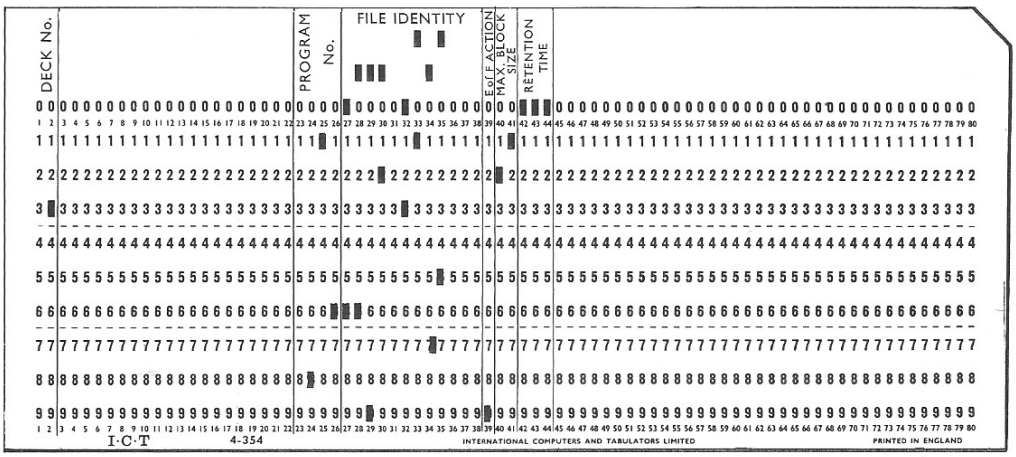
		RRN 80	RRN 86
ASSEMBLY	$\frac{1}{4}$"	551	588
	1" or $\frac{1}{2}$"	557	597
Object Program Running - all systems		141	133

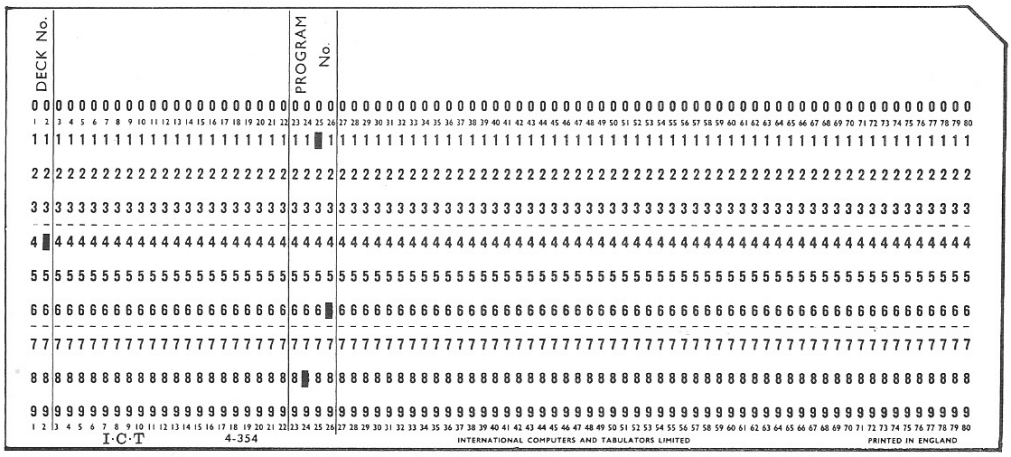
The Job Set-up cards needed for assembly are shown at the end of this chapter. The fields are punched as detailed in the 1300 Series Tape Housekeeping manuals.

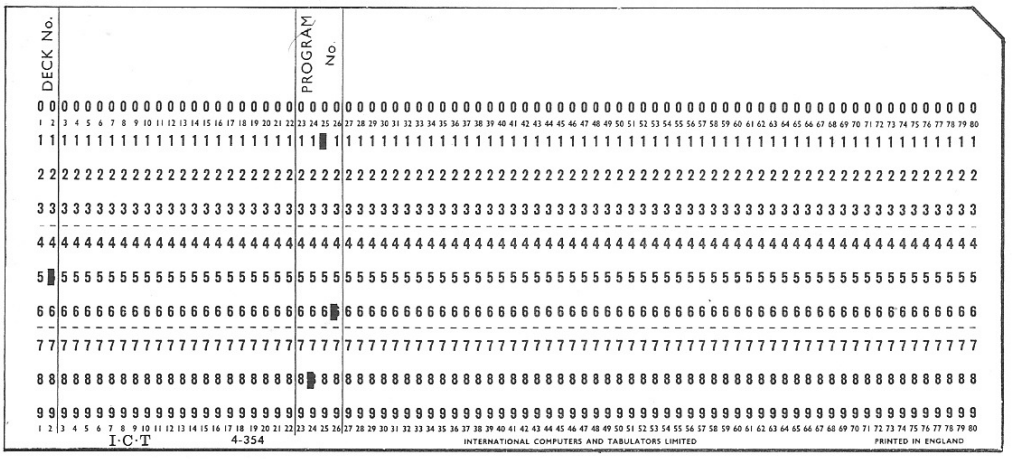

SEQ. No.
PROGRAM No.
DATE
PERIOD No. P. DECK 1
PERIOD No. P. DECK 2
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND
Repunch
0

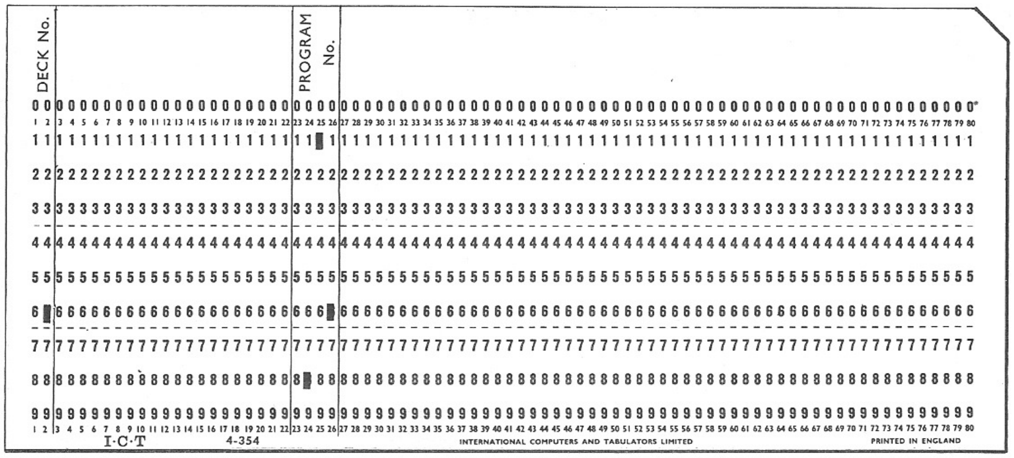

DECK No.
DRUM ADD RRN 80
DRUM ADD RRN 86
I.A.S. SIZE
DUMP REQ.
PROGRAM No.
FILE IDENTITY
E of F ACTION
MAX. BLOCK SIZE
RETENTION TIME
FILE IDENTITY of PROGRAM TAPE
PERIOD No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND
Repunch
I.C.T. MPL 2
Repunch
I.C.T. MPL 2

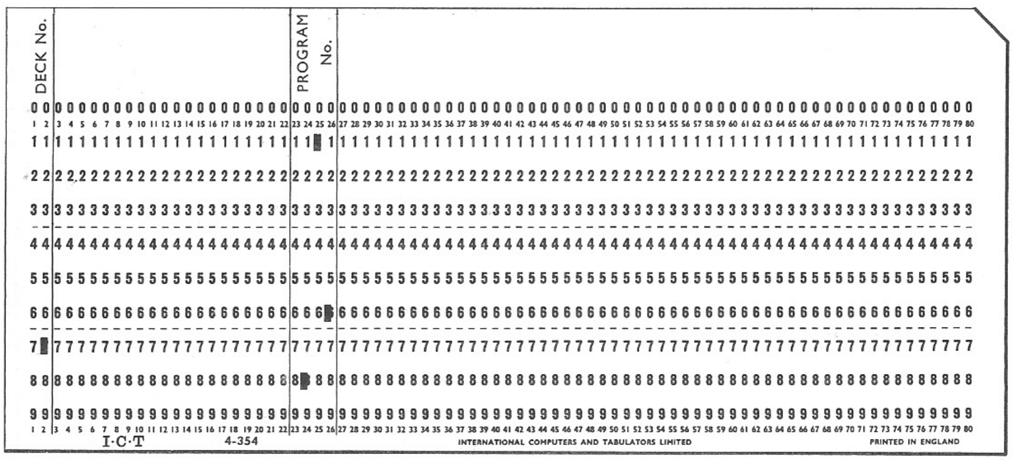

DECK No.
PROGRAM No.
FILE IDENTITY
E of F ACTION
MAX. BLOCK SIZE
RETENTION TIME
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND

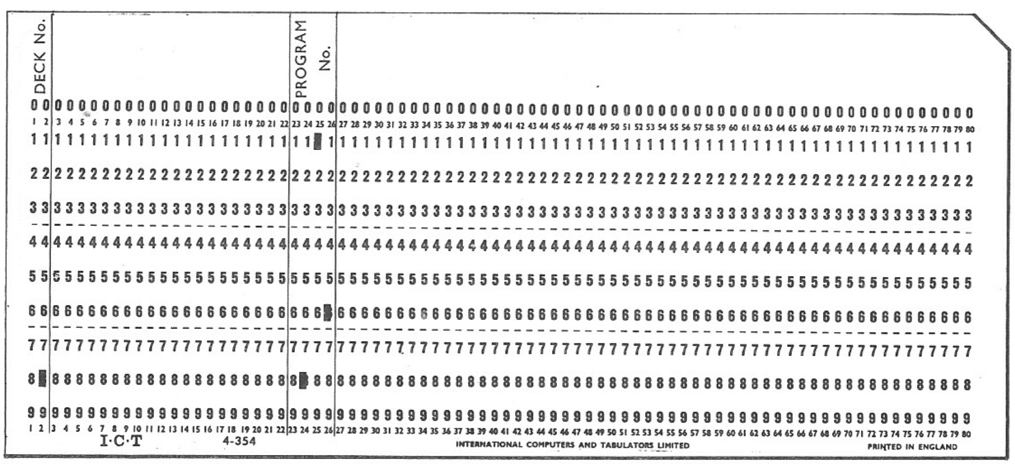

DECK No.
PROGRAM No.
FILE IDENTITY
E of F ACTION
MAX. BLOCK SIZE
RETENTION TIME
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND

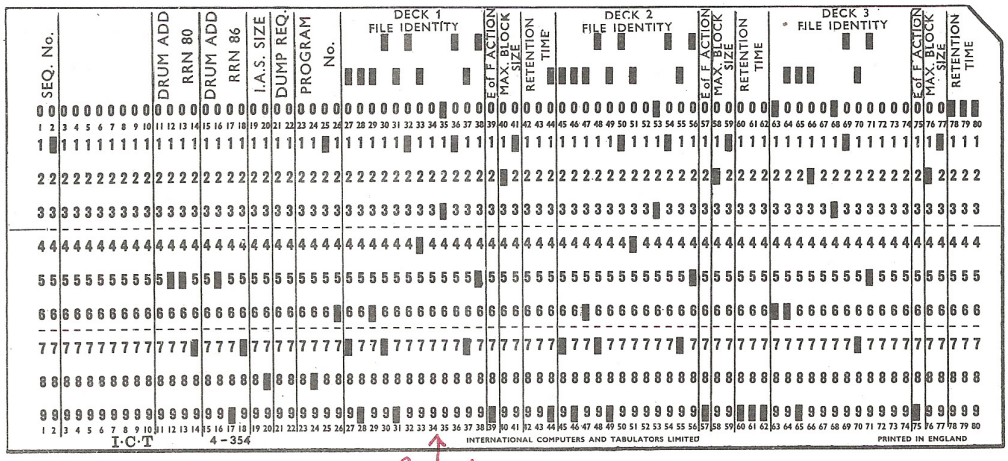

DECK No.
PROGRAM No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND


DECK No.
PROGRAM No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND


DECK No.
PROGRAM No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND

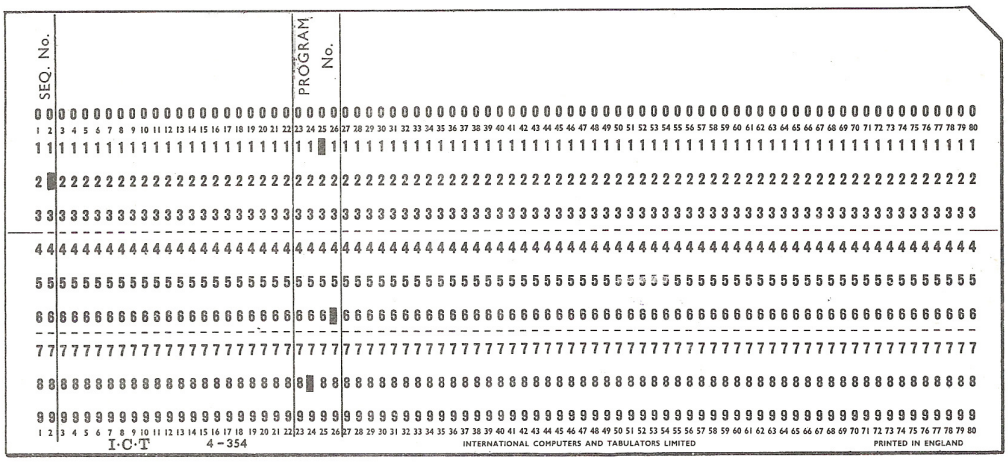

DECK No.
PROGRAM No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND

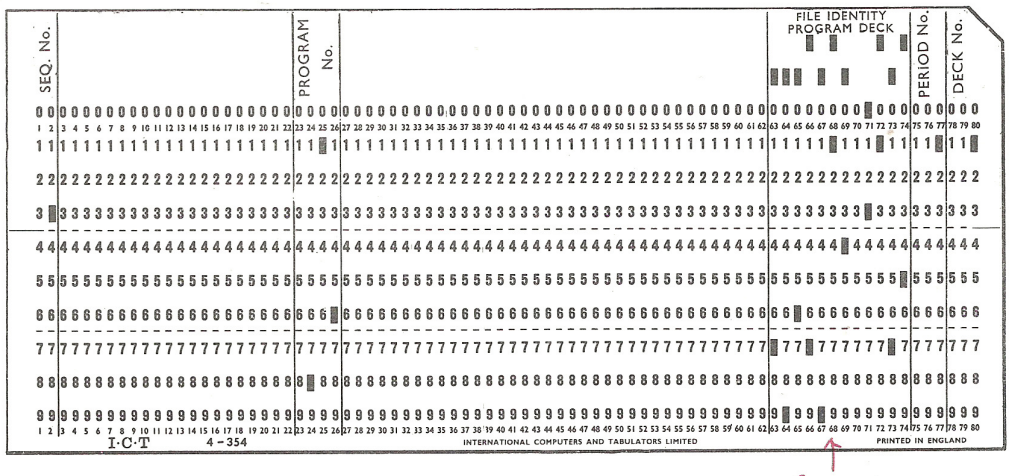

DECK No.
PROGRAM No.
I·C·T
4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND



Repunch

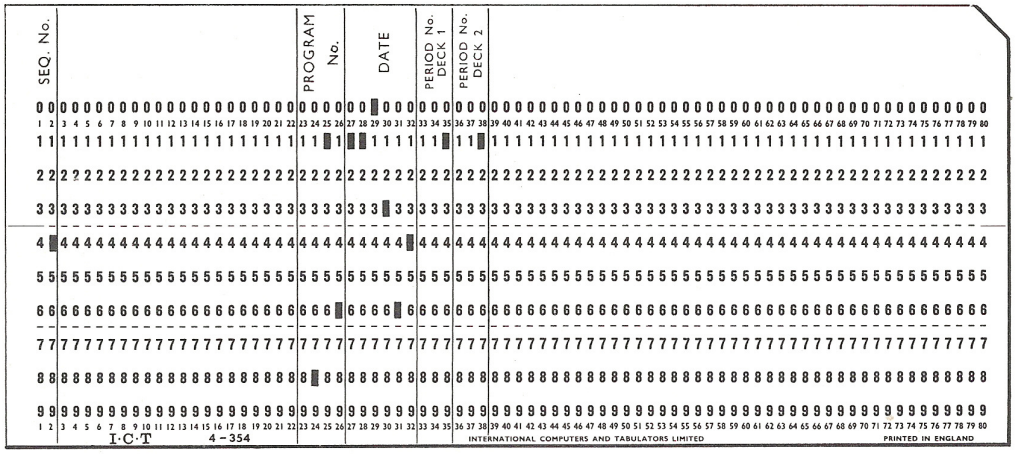
I.C.T MPh 2





Repunch

I.C.T. MPh 2


SEQ. No.
PROGRAM No.
DATE
PERIOD No. DECK 1
PERIOD No. DECK 2
I·C·T 4-354
INTERNATIONAL COMPUTERS AND TABULATORS LIMITED
PRINTED IN ENGLAND

Chapter 8

MPL 2 PROGRAM TAPE MAINTENANCE

This program will update old program tapes with newly assembled MPL 2 programs and it will also allow corrections and expansions to tape programs by means of standard program cards and will write card programs to tape. The system assumes that both input tapes and control cards are correctly sequenced, i.e. in program number order. THIS SYSTEM OF PROGRAM TAPE MAINTENANCE APPLIES TO ABSOLUTE MACHINE CODE OBJECT PROGRAMS AND NOT MPL2 SOURCE PROGRAMS

BASIC OPERATION

The old program on deck 3 is written to the drum and is overwritten by the new program on deck 4, the length of the updated program being the combined length of the two programs. If the control cards contain a program number which does not exist on tape, then this card program will be inserted in the new master program tape in the appropriate position.

Control cards can delete programs from the old master program tape or allow amendments to either tape (control cards have top priority). These cards use the MPL 2 field layout and the last card designation used by MPL 2 data cards. Unpunched MPL 2 card columns will be ignored in the same way as in the assembly.

DELETE

This statement only deletes programs from the old tape. Thus, if it is required to replace one program by another of the same program number, the number on the old master tape must be deleted.

Delete occupies one field with a right-justified program number (see Summary *1 on page 95).

F	OPERAND
-	- N N N N

All the sectors of an MPL 2 object program can be deleted by punching only the MPL 2 program number and leaving the sector number unpunched, i.e.

F	OPERAND
-	- N N - -

If no program is found corresponding to a delete statement, the field will be ignored and a line will be printed to this effect.

MODIFY AND INSERT PROGRAM ON CARDS

This function has a function digit of 'C' (Cards) and the program number right-justified in the rest of the block.

F	OPERAND
C	- N N N N

If the program is found on the old master program tape only, this program will be loaded on the drum. If this program is also on the new object program tape, then this latter program is the one which is updated. The amendments specified in the next two fields of the card will then be made. The second field must contain the number of consecutive words that the amended or inserted program will occupy on the drum. If this number is the same as that previously held on tape, the second field on the card must contain STET.

The third field should contain an 'E' function digit and if the program is already on tape and the original 'E' word is to be retained, the rest of this field should contain the starting decade address on the drum, unless this too is to be retained, in which case STET must be punched in the field.

The whole control macro is:

F	OPERAND
C	- P P S S

F	OPERAND
-	n n n n n
-	S T E T -

F	OPERAND
E	- D D D D
-	S T E T -

E = 'E' card from ~~tape~~ CARDS

DDDD = new start address from cards.

There may be more control fields on the same card but there might not be any more control cards preceding the amendment (insertion cards). Any standard or fast-read program cards can be read but they must be determined by an 'E' card. This 'E' card may not be valid (i.e. usable). If the programmer has specified that the entry condition on the object tape is obtained from card then this 'E' word will be checked for validity. The old 'E' word will be used in error cases. Entries to sectors can be amended similarly. Any errors will result in a line of print.

If an unset relativizer or a sequence error is detected, the remainder of these amendment cards will be ignored. The program will only be written to the new tape (in partially amended form) if it was on the old master program tape. If an attempt is made to read program to a drum area before the specified drum start address, this amendment will be ignored and a line will be printed.

LIMITATION

It is not possible to write programs to tape if the programs are larger than the drum size minus 1400 words; i.e. for a 12,000-word drum, the program must not exceed 10,600 words.

The MPL 2 Program Tape Maintenance will cope with multi-reel master files, but it is strongly recommended that this facility should not be used because of the problems of inter tape sector jump instructions.

OPERATING

Indicator 20 can be used to exclude any card reading. Indicator 21 will ignore the old master program tape; that is, amendments can be made to partially tested tape without reassembly. Indicator 22 will write programs to the master file from cards using 'C' designation control cards.

3139(8.64)

Note

Block sizes must not exceed 202 words for any tape to be used in this process.

SUMMARY

Program Deck	Tape	Writing Ring Required	Deck not required if indicator set
1	MPL 2 Assembler	No	must always be present
2	New Object Program Tape	No	MI 22
3	Old Master Program Tape	No	MI 21 or 22
4 (*2)	New Master Program Tape	Yes	must always be present
Cards			MI 20

Delete

-	-	P	P	S	S

Delete Sector

or

-	-	P	P	-	-

Delete all sectors of a program

Insert or Amend

C	-	P	P	S	S

-	n	n	n	n	n
-	S	T	E	T	-

E	-	D	D	D	D
-	S	T	E	T	-

Last Field

$\overline{13}$	$\overline{13}$	$\overline{13}$	$\overline{13}$	$\overline{13}$	$\overline{13}$

*1
- PP = program number
- SS = sector number
- nnnnn = number of consecutive words in program if different from that on tape
- E = 'E' card as ~~on tape~~ FOR CARDS
- DDDD = decade start of program on drum.

*2 If only three tapes are to be used, the MPL 2 master program tape can be unloaded after the program is on the drum and the deck can be used for any other tape.

Job Set-up

The MPL 2 assembly program tape should be loaded on deck 1, program number 826.

RRN 80 is 20.

RRN 86 is 57.

I.A.S. size is 8.

Dump is blank .

Appendix A

MPL 2 FUNCTIONS

Function No.	Function	Equivalent Machine Function Code	Page No.
A	Add To	64, 74	
B	Block Move	45	
C	Clear Add	60, 70	
D	Drum Reference	80, 81	
E	Enter Machine Code		
F	Fetch	37	
G	Generate Machine Function		
H	Isolate Digits		
I	Indicators	8, 9*	
J	Jump on Indicator	4*	
K	Compare	68, 78	
L	Label		
M	Mask	35	
N	Negate	61, 71	
⊖	Or (logical function)	36	
P	Print		
Q	Punch		
R	Read		
S	Subtract From	65, 75	
T	Tally	67, 77	
U	Unit Increase	66, 76	
V	Values to be Set		
W	Write Away	42	
X	Multiply	69, 79	
Y	1300 Series Functions		
Z	Zeroize	40, 57	
&	Plus	62, 72	
⊖	Minus	63, 73	
/	Divide		
⊡	Stop	11	
¼	Modify		
%	Define Data Name		
@	Store Link and Jump to Subroutine		
½	Tape Read		
¾	Tape Write		

* Equivalent machine designations

Appendix B

OPERATING INSTRUCTIONS FOR THE QUARTER INCH (16kc/s) TAPE SYSTEM

The operating instructions for the quarter inch (16 kc/s) tape system are shown on this page and the facing page.

Peripheral Units (Tick those used)	CARD PUNCH	PAPER TAPE PUNCH	TYPEWRITER	MAGNETIC TA
	If Indicator 24 set			Three decks

Special Instructions

1) Please write initial labels to program decks 2 and 3.
2) Line up paper to the beginning of page.
3) Sort Job Set-up parameter cards if necessary.

OP. SEQ. No.	MANUAL INDICATORS TO BE SET	MANUAL INSTRUCTIONS	EXPECTED EFFECT	STOP
1	21	I.O. and START	Reads job set-up E card	11 1111
2	-	START	Job set-up	11 1901
3	21, 22, 23, 24, 25, 26	START	MPL 2 Assembly	11 1329
	(See Chapter 7)			

ERROR STOP	OPERATOR ACTION	ERROR STOP	OPERATOR ACTION
11 1321	Abandon (Sequence error)	11 1328	Note stop and press start
11 1322	Misfeed action	11 1344	" " " " "
11 1323	Abandon (Too many data names)	11 1345	" " " " "
11 1340	Abandon (Invalid header card)		
11 1342	Abandon (Program too large)		

Print-out Requirements

IF FINAL STOP REACHED	IF UNEXPECTED STOP OR LOOPING
I.A.S. _____ Words	I.A.S. _____ Words
Drum _____ Channels at Decade _____	Drum _____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____

3139(8.64)

Magnetic Tape Loading Instructions

DECK ADDRESS	INITIAL SET-UP		SECOND SET-UP		THIRD SET-UP	
	TAPE REFERENCE No.	W.R.*	TAPE REFERENCE No.	W.R.*	TAPE REFERENCE No.	W.R.*
1	Assembly Tape					
2	Program Tape	W.R.				
3	Work Tape	W.R.				
4						
5						
6						
7						
8						

* Tick those tapes which require a writing ring.

For queued tapes put both tape reference numbers in the same box, following the second one by a '(Q)'.

Magnetic Tape Print-out Requirements (Tick where appropriate)

DECK ADDRESS	IF FINAL STOP REACHED					IF UNEXPECTED STOP OR LOOPING			
	Print to E/F Label	Print to Tapemark	Number of Blocks	Format		Print to Tapemark	Number of Blocks	Format	
				Full	Abbrev.			Full	Abbrev.
1									
2									
3									
4									
5									
6									
7									
8									

FORM 1/1742 (6.63) Printed in Great Britain by International Computers and Tabulators Limited London.

Appendix C

OPERATING INSTRUCTIONS FOR THE ONE INCH (90kc/s) AND HALF INCH (22½kc/s) TAPE SYSTEMS

The operating instructions for the one inch (90 kc/s) and half inch (22½ kc/s) tape systems are shown on this page and the facing page.

Peripheral Units (Tick those used)	CARD PUNCH	PAPER TAPE PUNCH	TYPEWRITER	MAGNETIC TA
	If Indicator 24 set			Three decks

Special Instructions

1) Please write initial labels to decks 2 and 3.

2) Line up paper to the beginning of page.

OP. SEQ. No.	MANUAL INDICATORS TO BE SET	MANUAL INSTRUCTIONS	EXPECTED EFFECT	STOP
1	21	I.O. and START	Reads job set-up E card	11 1111
2	-	START	Job set-up	11 0887
3	21, 22, 23, 24, 25, 26	START	MPL 2 assembly	11 1329
	(See Chapter 7)			

ERROR STOP	OPERATOR ACTION	ERROR STOP	OPERATOR ACTION
11 1321	Abandon (Sequence error)	11 1328	Note stop and press start
11 1322	Misfeed action	11 1341	" " " " "
11 1323	Abandon (Too many data names)	11 1345	" " " " "
11 1340	Abandon (Invalid header card)		
11 1342	Abandon (Program too large)		

Print-out Requirements

IF FINAL STOP REACHED	IF UNEXPECTED STOP OR LOOPING
I.A.S. _____ Words	I.A.S. _____ Words
Drum _____ Channels at Decade _____	Drum _____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____
_____ Channels at Decade _____	_____ Channels at Decade _____

3139(8.64)

Magnetic Tape Loading Instructions

DECK ADDRESS	INITIAL SET-UP		SECOND SET-UP		THIRD SET-UP	
	TAPE REFERENCE No.	W.R.*	TAPE REFERENCE No.	W.R.*	TAPE REFERENCE No.	W.R.*
1	Assembly Tape					
2	Program Tape	W.R.				
3	Work Tape	W.R.				
4						
5						
6						
7						
8						

* Tick those tapes which require a writing ring.

For queued tapes put both tape reference numbers in the same box, following the second one by a '(Q)'.

Magnetic Tape Print-out Requirements (Tick where appropriate)

DECK ADDRESS	IF FINAL STOP REACHED					IF UNEXPECTED STOP OR LOOPING			
	Print to E/F Label	Print to Tapemark	Number of Blocks	Format		Print to Tapemark	Number of Blocks	Format	
				Full	Abbrev.			Full	Abbrev.
1									
2									
3									
4									
5									
6									
7									
8									

FORM 1/1742 (6.63) Printed in Great Britain by International Computers and Tabulators Limited, London.

3139(8.64)

Appendix D

PUNCHING CODE

I.C.T. 5 ZONE

Zone	10	11	0	1
10	10			
11		11		
0			0	
1	A	J	&	1
2	B	K	S	%
3	C	L	T	¼
4	D	M	U	⊖
5	E	N	V	/
6	F	O	W	½
7	G	P	X	⊡
8	H	Q	Y	@
9	I	R	Z	¾

The following characters are used in source programs -

$\overline{12}$	8 and 4
$\overline{13}$	8 and 5
$\overline{14}$	8 and 6
$\overline{15}$	8 and 7
£	10, 0 and 6

Appendix E

MPL 2 SAMPLE PROGRAM

SHOPPING LIST GENERATOR

A simple domestic problem is used to illustrate stock control. A record exists for each item in the inventory and the inventory is updated daily and a printout is obtained of such items that have fallen below a specified minimum level. The amounts required to replenish the stock, the item and quantity cost and the total cost are also printed out at this stage. A periodic printout (in this example the period chosen is weekly i.e. each Saturday) is obtained stating the position of each stock item and the requirements, cost and total cost for bringing the store to its maximum holding.

The current level of the store cupboard is recorded on magnetic tape in blocks of five records, each nine words long. The format of a record is:

Word 1	Record length in digits 1 to 3
Word 2	Present level of stock (number of units)
Word 3	Optimum level of stock
Word 4	Minimum level of stock
Words 5 and 6	Zones and numerics of name of article. The numerics are used as a key to sort the records.
Words 7 and 8	Zones and numerics of units in which the article is measured.
Word 9	Price of one unit of the article.

Cards presented have a designation number in column 80. The first card is the date card with designation 5. The day of the week is punched in columns 1 to 12. If the day is a weekday (i.e. not Saturday), the cards which follow have a designation 2; these are updating cards and have the format:

Columns 1 to 12	Name of article
Columns 13 to 15	Change in stock level

If an item is removed from the store cupboard this negative quantity is represented by a $\overline{10}$ over -punching in column 13. The cards, sorted by the key word, are fed into the computer and files are updated. Should any of the alterations cause a stock level to fall below the minimum, the order quantity required to reach optimum level is calculated and a shopping list is printed out. A total cost is accumulated and printed out at the end of the run.

On a Saturday, several types of cards may be presented. These are:

Designation 1 (Column 80)		Insert a record
Columns	1 to 12	Name
	13 to 15	Optimum level
	16 to 18	Minimum level
	19 to 21	Present level
	22 to 33	Units
	40 to 44	Price

Designation 2 (Column 80)	Update a record.	(Format as for a weekdav.)
Designation 3 (Column 80)	Amend a record.	(Format as for inserting a record.)
Designation 4 (Column 80)	Delete a record.	
Columns 1 to 12	Name	

Every record on the file is examined and is brought up to optimum level.

The sample program described above is illustrated by the following Figures:

Figure 6	Flowchart
Figures 7(a) to 7(b)	MPL 2 Program
Figure 8	List of data names
Figure 9	Sample printout of MPL 2 program
Figure 10	Sample printout of data names
Figure 11	Sample printout of sector header card
Figure 12	Sample printout of daily statement
Figure 13	Printout of object program.

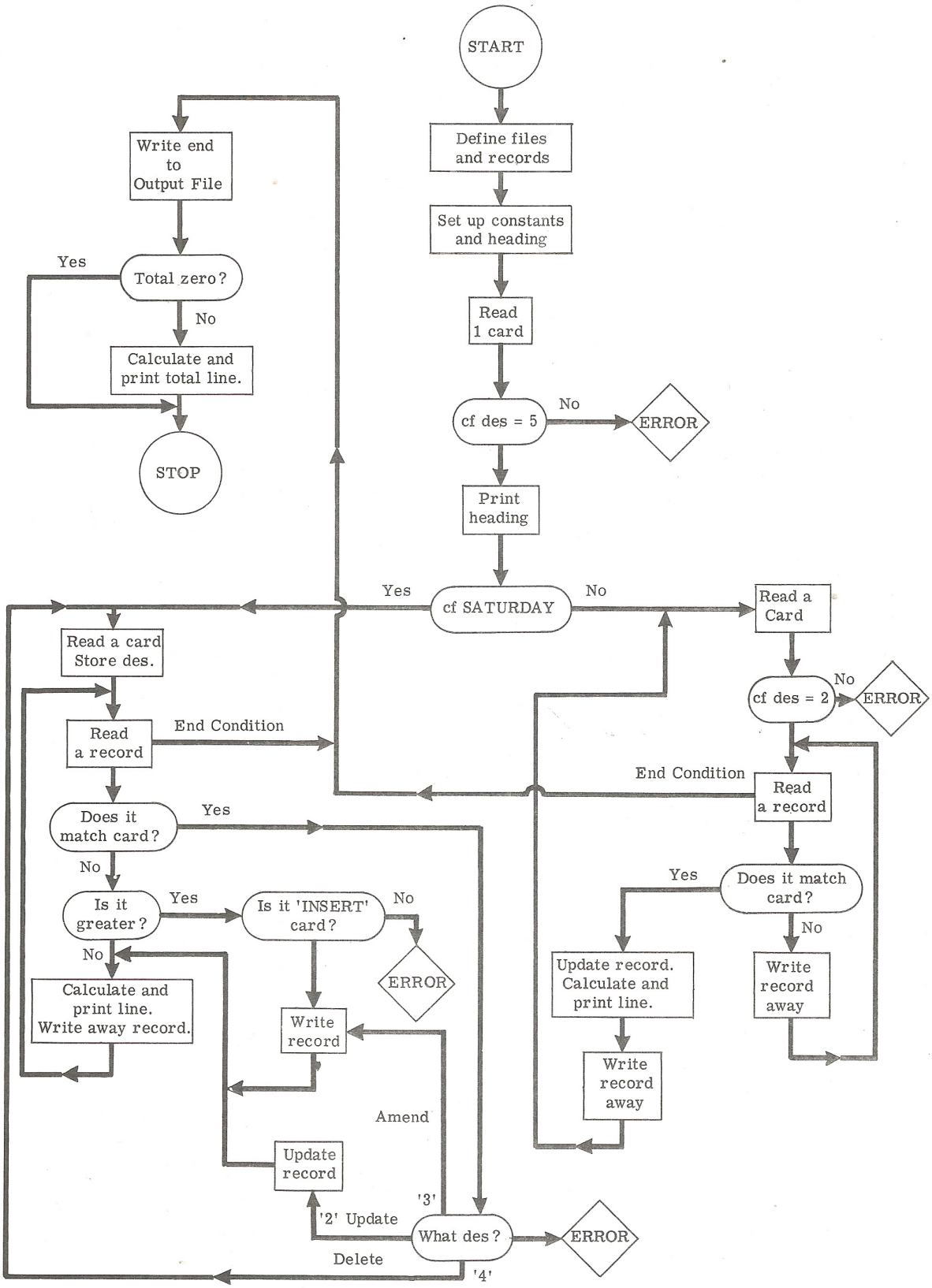


Figure 6: SHOPPING LIST GENERATOR FLOWCHART

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR — DATE 29 / 5 /64

PROGRAMMER D.D. — SHEET No. 1 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	0	1	L	0	1		0
			%	032	2	Define 2 files	
			O	FILEA	3	Input	
				046	4		
			9	FILEB	5	Output	
				046	6	Double Write Area	
			%	321	7		
				FILEA	8	Redefine file A as	
				REC A	9	record A	
				79	10		
			1	WORDS	11		
			1	PREST	12		
77	1	2	1	OPTIM	1		
			1	MINIM	2		
			2	NAMEA	3		
			2	UNITS	4		
			1	PRIC£	5		
			%	21	6	Define record C	
				REC C	7		
				79	8		
			1	WORDC	9		
			1	PRESC	10		
			1	OPTMC	11		
			1	MINMC	12		
77	2	3	2	NAMEC	1		
			2	UNITC	2		
			1	PRCC£	3		
			F	1	4	Set MOD = 1	
			W	MOD	5		
			%	1	6	Set values for	
			4	HEAD1	7	heading	
			V	4	8		
				HEAD1	9		
			4	23332	10		
			3	2 32	11		
			2	86779	12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(a): MPL 2 PROGRAM

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR	DATE 29 / 5 /64
PROGRAMMER D.D.	SHEET No. 2 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	3	4	5	739	1		
			4	4	2		
				0	3		
			23		4		
				0	5		
			L	1	6		1
			R	801	7	Read a card	
				99	8	go to 99 if last card.	
			K	5	9	cf des = 5	
			J	012	10		
			⊡	22220	11	Stop. Wrong designation	
			L	2	12		2
77	4	5	R	1 1	1	Read day card	
			O	DATE	2		
			O	0112	3		
			P	3122	4	Print heading	
			O	HEAD1	5		
				2014	6		
			O	DATE	7		
				4009	8		
			V	1	9	Set up numerics of	
				WS00	10	SATURDAY	
			2	13494	11		
			1	8	12		
77	5	6	¼	M~~0~~D	1		
			F	DATE	2		
			K	WS00	3		
			J	012 0	4	Go to 20 if it is SATURDAY	
			L	3	5		3
			R	801	6	Read a card	
				99	7		
			K	2	8	if designation = 2	
			J	012 4	9		
			⊡	22220	10	Stop. Wrong designation	
			L	4	11		4
			@	32	12	Go to Read Des 2 card S/R	

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(b)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR	DATE 29 / 5 /64
PROGRAMMER D. D.	SHEET No. 3/8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	6	7		LINK	1		
			L	5	2		5
			½	1A 88	3	Read record A	
				REC A	4		
			¼	MOD	5		
			F	NAME A	6		
			¼	MOD	7		
			K	NAME C	8	Compare key words	
			J	01 6	9	Jump if equal	
			¾	1B 5	10	Write record away	
				REC A	11		
			J	00 5	12	Loop to read next record	
77	7	8	L	6	1		6
			F	AMOUN	2	Update the record	
			A	PREST	3		
			F	MINIM	4		
			K	PREST	5		
			J	02 8	6	Jump if above minimum	
			@	40	7		
				LINK	8	Go to S/R to calculate and print	
			L	8	9		8
			¾	1B 5	10	Write record away	
				REC A	11		
			J	00 3	12	Go to read next card.	
77	8	9	L	99	1	Last card read	99
			½	1A 88	2	Read record A	
				REC A	3		
			¾	1B 5	4	Write record A	
				REC A	5		
			J	00 99	6	Loop until end of file	
			L	88	7	End of file	88
			¾	2B	8	Write end to file B	
			J	00 50	9	Jump to end.	
					10		
					11		
					12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(c)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR	DATE 29 / 5 /64
PROGRAMMER D.D.	SHEET No. 4 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	9	10	L	20	1	SATURDAY	20
			R	801	2	Read a card	
				98	3		
			W	DES	4	Store designation	
			R	1 1	5		
			O	NAMEC	6	Read name	
				112	7		
			L	21	8		21
			½	1A 87	9	Read a record	
				REC A	10		
			¼	MOD	11		
			F	NAM£A	12		
77	10	11	¼	MOD	1		
			K	NAMEC	2	Compare key words	
			J	01 22	3		
			J	03 23	4	If insert	
			J	00 24	5	Go to S/R	
			J	00 21	6	Return to read a record	
			L	22	7	Delete, update or amend	22
			F	DES	8		
			K	2	9		
			J	01 25	10	Jump if Update	
			K	3	11		
			J	01 26	12	Jump if Amend	
77	11	12	K	4	1		
			J	01 27	2	Jump if Delete	
			⊡	2220	3	Stop. Wrong designation	
			L	27	4	Delete	27
			J	00 20	5	Go to read next card	
			L	26	6	Amend	26
			J	00 33	7	Go to read card & write record	
			J	00 20	8	Go to read next card	
			L	25	9	Update	25
			@	32	10	Go to S/R	
				LINK	11		
			F	AMOUN	12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(d)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR — DATE 29 / 5 / 64

PROGRAMMER D. D. — SHEET No. 5 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
7,7	12	13	A	PRESET	1		
			J	00 24	2	Go to write record	
			J	00 20	3	Loop to read next card	
			L	23	4	Insert a record	23
			F	DES	5		
			K	1	6	cf designation = 1	
			J	01 28	7		
			⊡	22220	8	Stop. Wrong designation	
			L	28	9		28
			J	00 33	10	Go to read card	
			J	00 24	11	Go to write record	
			J	00 20	12	Loop to read next card	
7,7	13	14	L	33	1	Read card S/R	33
			Y	41	2		
				WS11	3	Set link in WS 11	
			R	1 5	4		
			1	~~0~~PTMC	5		
			O	133	6		
			1	MINMC	7		
			O	163	8		
			1	PRESC	9		
			O	193	10		
			O	UNITC	11		
			O	2212	12		
7,7	14	15	$\overline{10}$	PRCC£	1		
			O	405	2		
			V	1	3	Set up size of record	
				WORDC	4	in first word	
				9	5		
				0	6		
			¾	1B 5	7	Write record C away	
				REC C	8		
			Y	004	9	Jump to link	
				WS11	10		
					11		
					12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(e)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR — DATE 29 / 5 /64

PROGRAMMER D.D. — SHEET No. 6 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	15	16	L	24	1	Write record S/R	24
			Y	41	2	Store link	
				WS11	3		
			@	40	4	Go to calculate and	
				LINK	5	print line.	
			¾	1B 5	6	Write away record A	
				REC A	7		
			Y	.004	8	Go to link.	
				WS11	9		
			L	98	10	Last card reached.	98
			½	1A 87	11	Read a record.	
				REC A	12		
77	16	17	J	00 24	1	Go to write away	
			J	00 98	2	Loop to read next record	
			L	87	3	End of file reached	87
			¾	2B	4	Write end to file B	
			J	00 50	5	Jump to end.	
					6		
					7		
					8		
					9		
					10		
					11		
					12		
					1		
					2		
					3		
					4		
					5		
					6		
					7		
					8		
					9		
					10		
					11		
					12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(f)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR							DATE 29 / 5 /64
PROGRAMMER D. D.							SHEET No. 7/8
PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
7,7	1,7	1,8	L	1 3,2	1	Read updating card S/R	32
			R	1 2	2		
			O	N,A,M,E,C	3		
				1,1,2	4		
			1	A,M,Θ,U,N	5		
				1,3,0,3,4	6		
			Y	0,0,4	7	Jump to link	
				L,I,N,K	8		
			L	4,0	9	Calculate and print	40
			F	Θ,P,T,I,M	10	line S/R.	
			⊖	P,R,E,S,T	11		
			W	Q,T,Y	12		
7,7	1,8	1,9	X	P,R,I,C,£	1	Calculate cost	
			W	C,Θ,S,T,£	2		
			J	0,1 4,1	3		
			V	2	4		
				W,S,0,1	5		
				0	6	} Set up '@'	
				5	7		
				0	8		
				8	9		
			P	3,0,3 6	10	Print line	
			O	N,A,M,E,A	11		
			O	2,0,1,2	12		
7,7	1,9	2,0	1	Q,T,Y	1		
			O	3,6,0,3,1	2		
			O	U,N,I,T,S	3		
			O	4,0,1,2	4		
			O	W,S,0,1	5		
			O	5,3,0,1	6		
			10	P,R,I,C,£	7		
			O	5,5,0,8,1	8		
			10	C,Θ,S,T,£	9		
			O	7,0,0,8,1	10		
			F	C,O,S,T,£	11		
			A	T,Θ,T,L,£	12	Accumulate total	

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(g)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES

PROGRAM SHEET

TITLE SHOPPING LIST GENERATOR | DATE 29 / 5 /64

PROGRAMMER D. D. | SHEET No. 8 / 8

PROG. No.	THIS CARD SEQUENCE NUMBER	NEXT CARD SEQUENCE NUMBER	F	OPERAND	LINE	NARRATIVE	LABEL
77	20	21	L	41	1		41
			Y	004	2	Jump to link.	
				LINK	3		
			L	50	4	End of run.	50
			C	TOTL£	5		
			J	01 51	6		
			V	4	7	Set up heading	
				HEAD2	8		
			4	3423	9		
				2432	10		
			3	6313	11		
				5775	12		
77	21		3	22443	1		
			2		2		
			5	49349	3		
			5		4		
			P	3062	5	Print total time	
			O	HEAD2	6		
			O	2019	7		
			$\bar{1}$0	TOTL£	8		
			O	70081	9		
			L	51	10		51
			⊡	13332	11	Stop. with runout	
			L	BE	12	End of program.	
					1		
					2		
					3		
					4		
					5		
					6		
					7		
					8		
					9		
					10		
					11		
					12		

FORM 1/1704 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 7(h)

I.C.T. MNEMONIC PROGRAMMING LANGUAGE - 1300 SERIES
RECORD OF DATA NAMES

TITLE *SHOPPING LIST GENERATOR* DATE 29 / 5 / 64

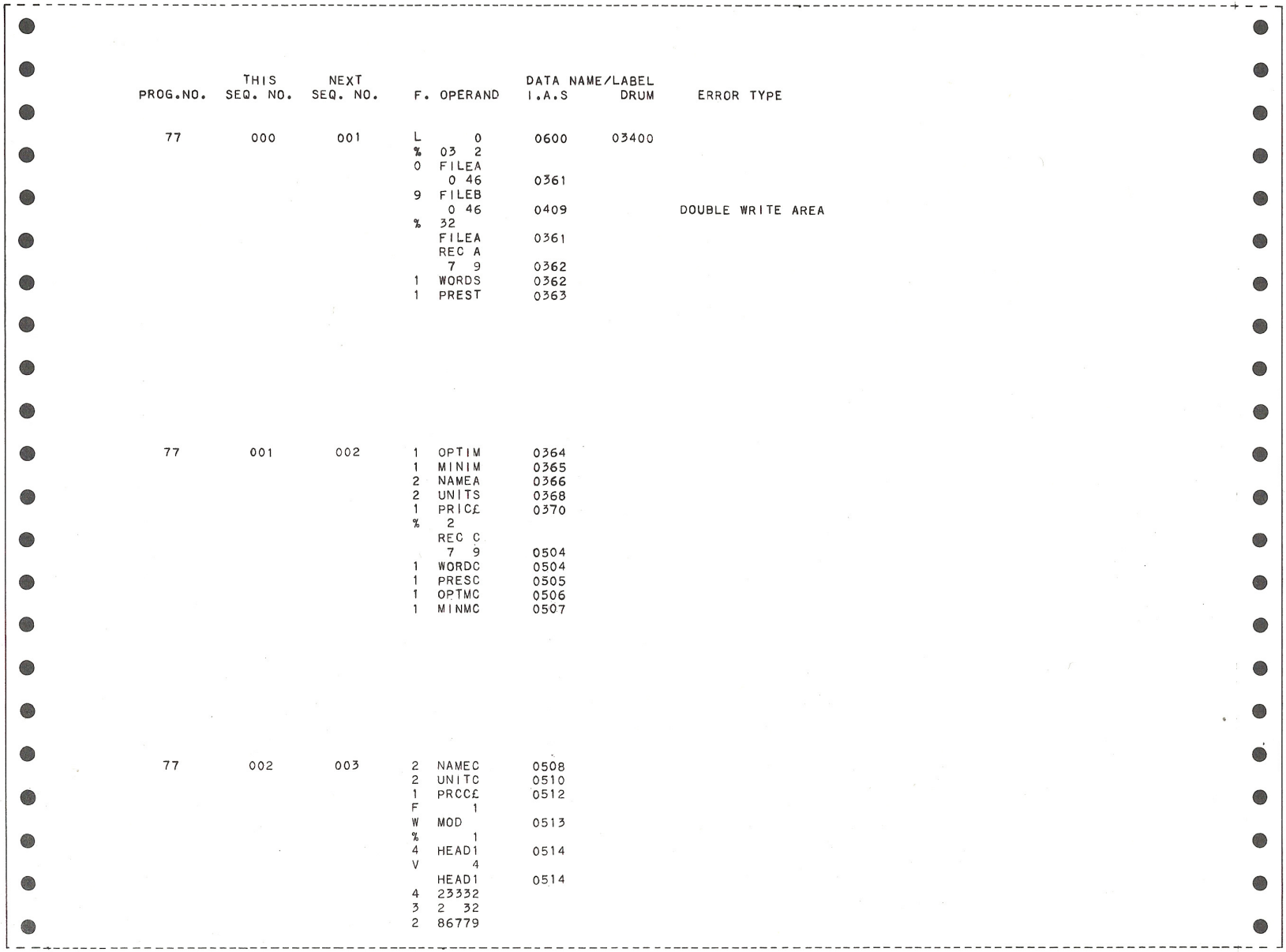
PROGRAMMER D. D. PROGRAM No. 77 SHEET No. 1/1

DATA NAME	REMARKS	No. OF WORDS	DATA NAME	REMARKS	No. OF WORDS
FILEA	Input	48			
FILEB	Output, double write area	48			
RECA	Redefines FILEA	9			
WORDS		1			
PREST		1			
OPTIM		1			
MININ	Redefine REC A	1			
NAMEA		2			
UNITS		2			
PRIC£		1			
RECC		9			
WORDC		1			
PRESC		1			
OPTMC	Redefine REC C	1			
MINMC		1			
NAMEC		2			
UNITC		2			
PRCC£		1			
MOD	=1	1			
HEAD1	SHOPPING LIST	4			
DATE		1			
LINK	Link word	1			
AMOUN	Signed before	1			
DES		1			
QTY		1			
COST£		1			
TOTL£		1			
HEAD2	TOTAL EXPENDITURE	4			

FORM 1/1705 (7.63)

Printed in Great Britain by International Computers and Tabulators Limited, London.

Figure 8: LIST OF DATA NAMES



```
PROG.NO.  THIS SEQ. NO.  NEXT SEQ. NO.   F. OPERAND      DATA NAME/LABEL I.A.S   DRUM     ERROR TYPE

   77          000            001        L      0          0600        03400
                                         %  03  2
                                         0  FILEA
                                            0 46          0361
                                         9  FILEB
                                            0 46          0409                 DOUBLE WRITE AREA
                                         %  32
                                            FILEA         0361
                                            REC A
                                             7  9         0362
                                         1  WORDS         0362
                                         1  PREST         0363

   77          001            002        1  OPTIM         0364
                                         1  MINIM         0365
                                         2  NAMEA         0366
                                         2  UNITS         0368
                                         1  PRIC£         0370
                                         %   2
                                            REC C
                                             7  9         0504
                                         1  WORDC         0504
                                         1  PRESC         0505
                                         1  OPTMC         0506
                                         1  MINMC         0507

   77          002            003        2  NAMEC         0508
                                         2  UNITC         0510
                                         1  PRCC£         0512
                                         F      1
                                         W  MOD           0513
                                         %      1
                                         4  HEAD1         0514
                                         V      4
                                            HEAD1         0514
                                         4  23332
                                         3  2  32
                                         2  86779
```

Figure 9: SAMPLE PRINTOUT OF AN MPL2 PROGRAM

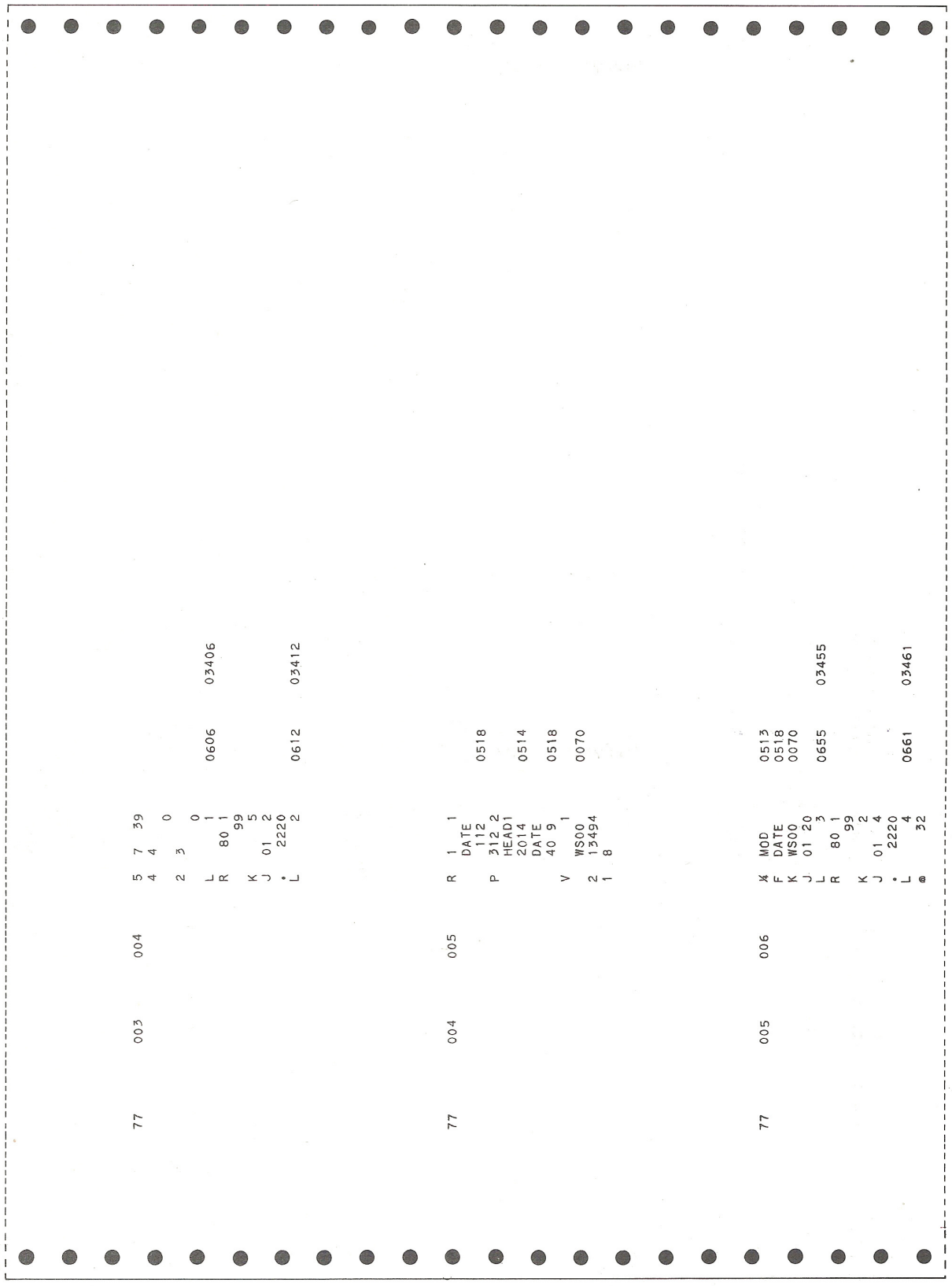


Figure 9 Continued

3139(8.64)

DATA NAME	SIZE	ADDRESS	DATA NAME	SIZE	ADDRESS
HEAD1	004	0514	HEAD2	001	0526
DES	001	0522	COST	001	0524
REC A	009	0362	REC C	009	0504
PRCC	001	0512	TOTL	001	0525
OPTMC	001	0506	FILEA	047	0361
FILEB	047	0409	DATE	002	0518
OPTIM	001	0364	MOD	001	0513
NAMEA	002	0366	NAMEC	002	0508
LINK	001	0520	PREST	001	0363
PRESC	001	0505	MINMC	001	0507
MINIM	001	0365	AMOUN	001	0521
QTY	001	0523	UNITS	002	0368
UNITC	002	0510	PRIC	001	0370
WORDS	001	0362	WORDC	001	0504

Figure 10: SAMPLE PRINTOUT OF DATA NAMES

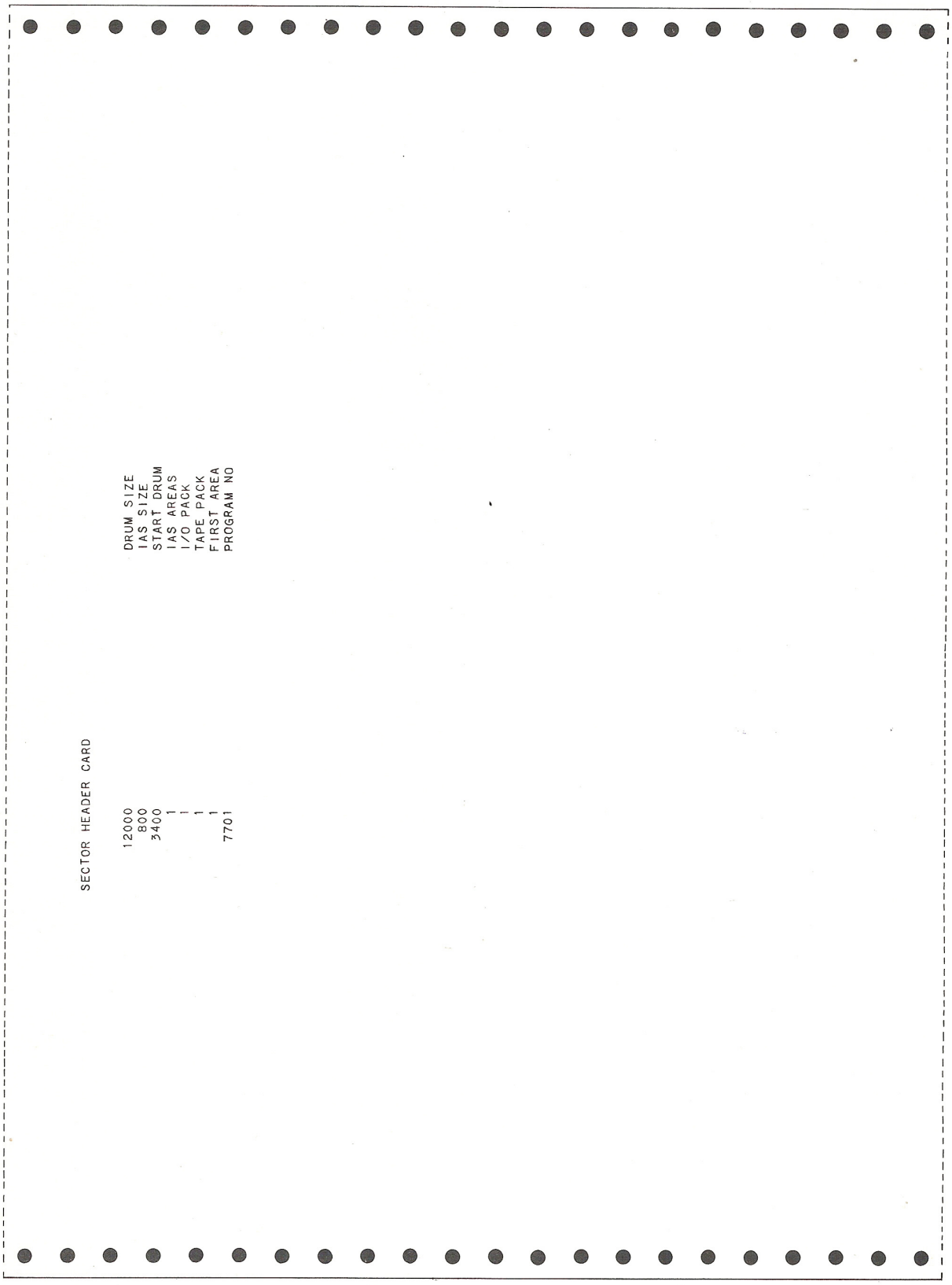
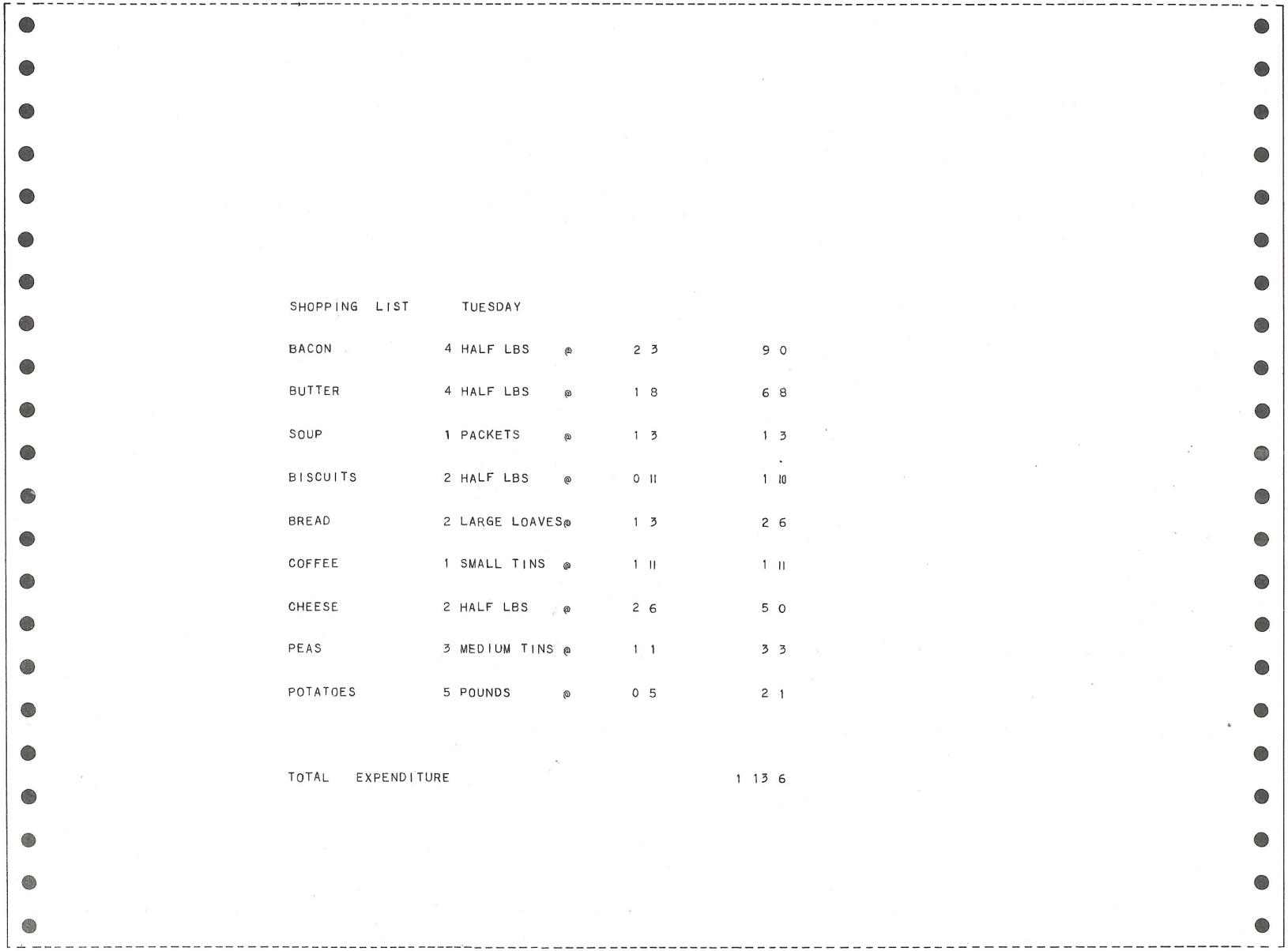


Figure 11: SAMPLE PRINTOUT OF SECTOR HEADER CARD

Figure 12: SAMPLE PRINTOUT OF DAILY STATEMENT



SHOPPING LIST	TUESDAY			
BACON	4 HALF LBS	@	2 3	9 0
BUTTER	4 HALF LBS	@	1 8	6 8
SOUP	1 PACKETS	@	1 3	1 3
BISCUITS	2 HALF LBS	@	0 11	1 10
BREAD	2 LARGE LOAVES	@	1 3	2 6
COFFEE	1 SMALL TINS	@	1 11	1 11
CHEESE	2 HALF LBS	@	2 6	5 0
PEAS	3 MEDIUM TINS	@	1 1	3 3
POTATOES	5 POUNDS	@	0 5	2 1
TOTAL EXPENDITURE				1 13 6

OBJECT PROGRAM

03400	00	40	80	120	160
0600	45 0799	54 0009	00 0004	30 0779	00 0004
	01 0315	36 0106	00 4115	41 0520	00 4115
0601	45 0798	37 0518	00 4655	00 4775	00 4081
	01 0408	57 0011	00 4682	00 4722	00 4762
0602	45 0798	36 0094	37 0785	37 0521	37 0785
	01 0455	37 0518	00 4115	64 0363	00 4115
0603	45 0797	55 0001	01 4686	00 4756	01 4765
	01 0316	54 0010	37 0784	00 4688	00 4756
0604	37 0796	36 0095	00 0004	37 0522	00 4762
	42 0513	37 0519	00 4115	68 0796	00 4765
0605	45 0792	57 0011	00 4682	01 4726	37 0782
	04 0514	36 0106	00 4686	11 2220	00 4115
0606	00 4010	37 0519	37 0782	00 4728	00 4773
	37 0091	55 0001	00 4115	00 4756	00 0000
0607	68 0791	54 0010	00 4773	00 4688	00 0000
	01 4682	36 0107	00 4688	00 4728	00 0000
0608	37 0103	00 4000	00 4010	41 0081	00 0000
	54 0011	00 4649	37 0091	37 0092	00 0000
0609	57 0011	45 0788	68 0791	55 0010	00 0000
	42 0033	01 0070	01 4762	57 0009	00 0000
0610	68 0790	42 0064	37 0103	42 0506	00 0000
	01 4612	00 4651	54 0011	37 0093	00 0000
0611	11 2220	37 0513	57 0011	54 0006	00 0000
	00 4612	54 0006	42 0033	57 0011	00 0000
0612	37 0090	62 0653	42 0522	36 0506	00 0000
	57 0005	00 4060	37 0090	37 0093	00 0000
0613	55 0010	37 0518	57 0006	54 0007	00 4036
	42 0518	68 0070	55 0010	57 0009	10 3682
0614	37 0090	01 4688	42 0508	42 0507	00 4036
	55 0010	00 4655	37 0090	37 0093	10 3620
0615	42 0519	00 4010	55 0010	55 0010	00 4036
	37 0091	37 0091	42 0509	57 0009	10 3600
0616	57 0006	68 0791	37 0091	42 0505	00 4036
	54 0004	01 4682	57 0006	37 0094	10 3559
0617	36 0518	37 0103	54 0004	54 0006	00 0002
	37 0091	54 0011	36 0508	57 0011	00 0504
0618	55 0006	57 0011	37 0091	36 0505	00 9000
	57 0002	42 0033	55 0006	37 0094	00 0000
0619	36 0519	68 0787	57 0002	57 0006	00 4036
	37 0092	01 4661	36 0509	55 0007	10 3522
0620	57 0008	11 2220	37 0092	42 0510	00 0000
	36 0518	00 4661	57 0008	37 0094	00 0004
0621	37 0092	30 0786	36 0508	55 0007	00 0000
	54 0006	41 0520	37 0092	42 0511	00 0003
0622	57 0008	00 4775	54 0006	37 0095	00 0002
	36 0519	00 4663	57 0008	57 0006	00 0000
0623	37 0789	37 0785	36 0509	54 0001	00 4036
	42 0090	00 4115	00 4704	36 0510	10 3479
0624	40 0091	01 4686	37 0785	37 0095	00 0002
	00 4625	42 0064	00 4115	55 0006	00 0362
0625	45 0091	37 0513	01 4765	57 0005	00 0001
	20 0092	54 0006	42 0064	36 0511	00 0362
0626	45 0111	62 0667	37 0513	37 0096	00 4036
	03 0112	00 4060	54 0006	57 0011	10 3463
0627	37 0514	37 0366	62 0708	36 0510	00 0000
	57 0011	42 0064	00 4060	37 0096	00 0002

Figure 13: DRUM PRINTOUT OF OBJECT PROGRAM

3139(8.64)

0628	36 0092	37 0513	37 0366	54 0006	21 3494
	37 0514	54 0006	42 0064	57 0011	18 0000
0629	55 0001	62 0670	37 0513	36 0511	00 0000
	57 0002	00 4060	54 0006	37 0097	00 0012
0630	36 0093	68 0508	62 0711	54 0007	00 0000
	37 0514	01 4674	00 4060	57 0007	00 0005
0631	55 0011	37 0784	68 0508	42 0512	11 1111
	54 0010	00 4672	01 4714	00 4752	JJ JJJJ
0632	36 0094	00 0004	03 4724	45 0778	42 3332
	37 0515	00 4115	00 4756	01 0504	32 0032
0633	57 0011	00 4663	00 4704	37 0777	28 6779
	36 0104	00 4674	00 4714	00 4754	57 0039
0634	37 0515	37 0521	37 0522	00 0004	44 0000
	55 0001	64 0363	68 0787	00 4115	00 0000
0635	57 0002	37 0365	01 4720	00 4081	23 0000
	36 0105	68 0363	68 0781	00 4756	00 0000
0636	37 0515	02 4679	01 4719	41 0081	00 0000
	55 0011	00 4677	68 0780	00 4757	00 0001
0637	54 0010	30 0783	01 4718	30 0776	90 0409
	36 0106	41 0520	11 2220	41 0520	90 0456
0638	37 0516	00 4774	00 4688	00 4774	00 0046
	54 0009	00 4679	00 4719	00 4759	00 0001
0639	36 0094	37 0784	00 4728	37 0734	00 0361
	37 0517	00 4680	00 4688	00 4760	00 0001

Figure 13 Continued

03600	00	40	80	120	160
0600	37 0090	42 0792	74 0525	00 0000	36 0791
	57 0006	00 4753	00 4681	00 0000	67 0793
0601	55 0010	37 0791	00 4520	00 0000	03 4762
	42 0508	55 0009	00 4682	00 0000	33 0791
0602	37 0090	54 0005	70 0525	00 0000	36 0791
	55 0010	36 0094	01 4713	00 0000	00 4795
0603	42 0509	37 0792	45 0749	00 0000	41 0785
	37 0091	55 0009	04 0526	00 0000	37 0792
0604	57 0006	54 0005	37 0748	00 0000	55 0009
	54 0004	36 0106	42 0090	00 0000	42 0794
0605	36 0508	37 0368	40 0091	00 0000	37 0792
	37 0091	57 0011	00 4686	00 0000	57 0003
0606	55 0006	36 0094	45 0091	00 0000	42 0792
	57 0002	37 0368	20 0092	00 0000	00 4753
0607	36 0509	55 0001	45 0111	00 0000	55 0005
	37 0092	57 0002	03 0112	00 0000	42 0791
0608	57 0008	36 0095	37 0526	00 0000	37 0788
	36 0508	37 0368	57 0011	00 0000	36 0791
0609	37 0092	55 0011	36 0092	00 0000	37 0792
	54 0006	54 0010	37 0526	00 0000	54 0005
0610	57 0008	36 0096	55 0001	00 0000	42 0792
	36 0509	37 0369	57 0002	00 0000	37 0794
0611	37 0092	57 0011	36 0093	00 0000	54 0001
	55 0010	36 0106	37 0526	00 0000	57 0011
0612	57 0009	37 0369	55 0011	00 0000	54 0002
	42 0521	55 0001	54 0010	00 0000	36 0792

0613	37 0093	57 0002	36 0094	00 0000	37 0794
	54 0006	36 0107	37 0527	00 0000	04 4774
0614	57 0011	37 0369	57 0011	00 0000	35 0754
	36 0521	55 0011	36 0104	00 0000	62 0789
0615	37 0092	54 0010	37 0527	00 0000	04 4786
	54 0004	36 0108	55 0001	00 0000	66 0793
0616	57 0011	37 0071	57 0002	00 0000	57 0012
	68 0799	54 0008	36 0105	00 0000	01 4778
0617	02 4619	36 0096	37 0527	00 0000	37 0789
	03 4619	37 0072	55 0011	00 0000	36 0791
0618	61 0521	54 0008	54 0010	00 0000	37 0794
	42 0521	36 0108	36 0106	00 0000	54 0002
0619	00 4520	40 0793	37 0528	00 0000	57 0011
	00 4620	37 0370	54 0009	00 0000	42 0794
0620	37 0364	42 0792	36 0094	00 0000	60 0794
	63 0363	00 4763	37 0529	00 0000	68 0794
0621	42 0523	37 0791	54 0009	00 0000	01 4784
	79 0370	55 0004	36 0106	00 0000	42 0794
0622	42 0524	57 0006	40 0793	00 0000	60 0790
	01 4681	36 0096	37 0525	00 0000	63 0794
0623	45 0797	37 0791	42 0792	00 0000	64 0791
	02 0071	55 0010	00 4763	00 0000	40 0794
0624	37 0796	54 0010	37 0791	00 0000	37 0794
	42 0090	36 0097	55 0004	00 0000	36 0792
0625	40 0091	37 0792	57 0011	00 0000	65 5554
	00 4626	55 0004	36 0097	00 0000	65 5554
0626	45 0091	57 0006	37 0791	00 0000	57 0008
	20 0092	36 0108	55 0005	00 0000	64 0791
0627	45 0111	37 0792	54 0010	00 0000	36 0792
	03 0112	55 0010	36 0098	00 0000	00 477B
0628	37 0366	54 0010	37 0792	00 0000	00 0000
	57 0011	36 0109	55 0004	00 0006	00 0101
0629	36 0092	40 0793	57 0011	43 4230	00 0000
	37 0366	37 0524	36 0109	00 2432	00 4000
0630	55 0001	42 0792	37 0792	36 3130	00 0000
	57 0002	00 4763	55 0005	00 5775	00 000A
0631	36 0093	37 0791	54 0010	32 2443	5& &&&&
	37 0366	55 0004	36 0110	20 0000	5& &&&&
0632	55 0011	57 0011	00 4000	54 9349	5& &&&&
	54 0010	36 0097	00 4713	50 0000	5& &&&&
0633	36 0094	37 0791	00 4017	41 0795	5& &&&&
	37 0367	55 0005	11 3332	57 0012	5& &&&&
0634	57 0011	54 0010	00 0000	30 0789	5& &&&&
	36 0104	36 0098	00 0000	31 0792	5& &&&&
0635	37 0367	37 0792	00 0000	42 0791	36 0039
	55 0001	55 0004	00 0000	31 0791	00 4025
0636	57 0002	57 0011	00 0000	42 0791	00 0000
	36 0105	36 0109	00 0000	57 0008	00 0003
0637	37 0367	37 0792	00 0000	36 0791	00 0000
	55 0011	55 0005	00 0000	57 0004	00 0005
0638	54 0010	54 0010	00 0000	36 0791	00 0000
	36 0106	36 0110	00 0000	57 0002	00 0008
0639	40 0793	00 4000	00 0000	36 0791	00 0000
	37 0523	37 0524	00 0000	57 0001	00 0002

Figure 13 Continued