Documentation of programs.

1. Title - clear statement of what the program does, but concise enough for catalogue.
2. Type of program - linear, looping, branching (other than to terminate), linear array, subroutine, nested loops, string processing, 2-D array.
3. Description of program - amplification of title
4. Method of use
   a. Detailed instructions for data preparation
   Presentation of data, any restrictions on the range of data.
   b. Operating instructions
   The machine on which program runs + any other peripherals used should be stated. Also how to load program, run it + enter data.
   c. Results format
   Examples of the results layout should be given eg headings, tables etc.
   d. Error messages
   If any error messages are put in to vet input data (eg only values in range 1-10 are acceptable) these must be detailed. If there are none, state NONE.
5. Flowchart + Algorithm
   Neat and accurate representation of the algorithm. The correct boxes must be used, algorithm must be language independent ie. dont use BASIC statements.
6. Program
   a. Language
   b. Listing - produced by computer
7. Testing procedure
   a. Test data
   This should be designed to check all paths through the program.
   b. Expected results
   Show all working, formulae etc.
   c. Actual results
   These should be produced from the computer from the program and must be correct to gain marks.

15
20.

Problem

The customer accounts of a shop contain an account number, name, amount owing and credit limit. The credit limit is the largest amount the shop will allow the customer to owe them. A data card for 1 account would look like this:-

```
185  DATA 1416, " M.J. BROWN", 47.73, 60.00.
```

Write a program to read the data cards for ten accounts and print the information. If the amount owing is greater than the credit limit the words CREDIT LIMIT EXCEEDED are printed next to the information for that account.

Description:

The program contains example of four data cards containing records for 7 customers to a shop containing an account number, name, amount owing and credit limit. The program prints out the 7 data cards and if the credit limit is exceeded then the words CREDIT LIMIT EXCEEDED will be printed

Design of Input

This program uses READ and DATA statements to READ in the account number (A), name (N$), amount owing (O), and credit limit (C)

Formulae

The program checks to see if the credit is exceeded by using the IF and THEN statements as follows

```
IF O>C THEN PRINT " CREDIT LIMIT EXCEEDED."
```

Algorithm

1. Print Headings
2. Input Information
3. Check amount owing
4. Output result.

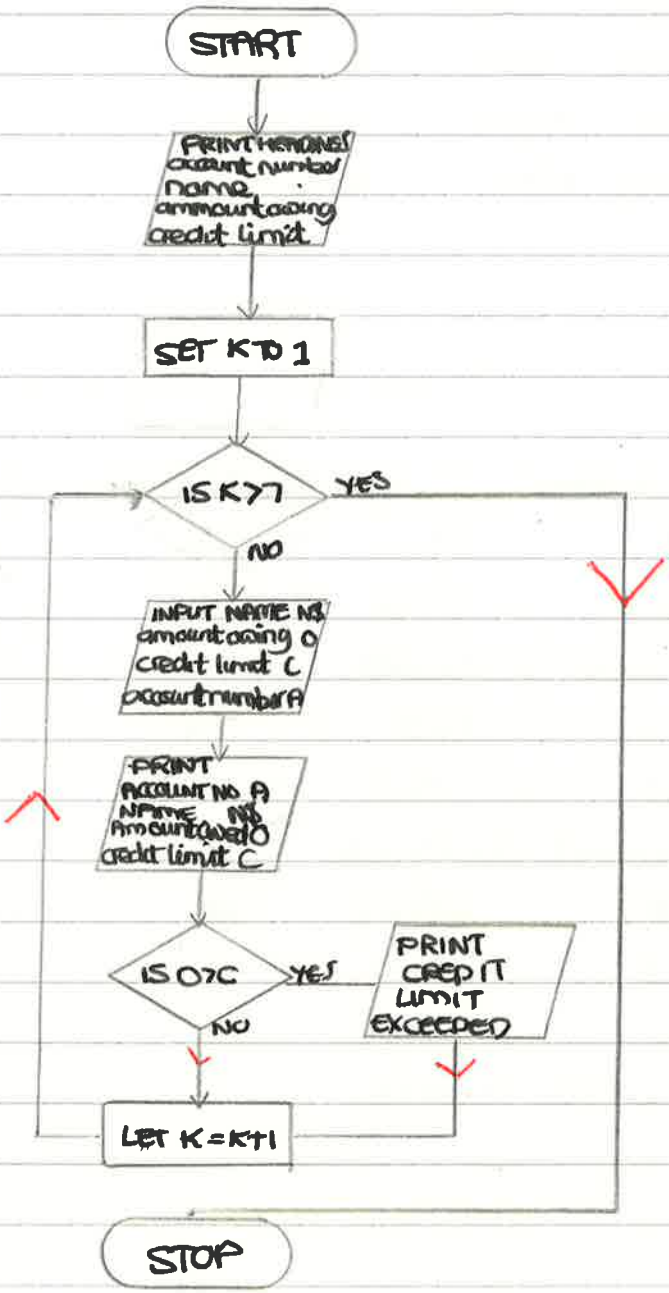
1st Refinement

1 Print Headings (account number, name, amount owing, credit limit).
2.1 Input account number
2.2 Input name
2.3 Input amount owing
2.4 Input credit limit
3. Check that amount owing is not greater than credit limit
4. Output Results - table with account number, name, amount owing, credit limit

2nd Refinement:

1. Print Headings (account number, name, amount owing, credit limit)
2.1 Input account number
2.2 Input name
2.3 Input amount owing
2.4 Input credit limit
3. Check that amount owing is not greater than the credit limit.
4. Output Results - table with account number, name, amount owing, credit limit.
4.1 If the amount owing than the credit limit print "CREDIT LIMIT EXCEEDED"

Flowchart



must have arrows in a loop

Programme

```
10 REM CREDIT
20 ? "Account Number * Name * Amount Owing * credit limit"
30 FOR K = 1 TO 7
40 READ A, N$, O, C                          * = Space.
50 ? A;"*********";N$;O;"*********" C
60 IF O>C THEN PRINT "CREDIT LIMIT EXCEEDED"
70 NEXT K
80 DATA 1000, " BILL", 36, 50
90 DATA 1001, " JOHN", 79, 65
100 DATA 1002, " FRED", 54, 60
```

```
110 DATA 1003, "MARK", 68, 65
120 DATA 1004, "JANE", 10, 40
130 DATA 1005, "ANDY", 18, 20
140 DATA 1006, "PAUL", 98, 50
150 END
```

Results

Account Number	Name	Amount Owing	Credit Limit
1000	BILL	36	50
1001	JOHN	79	65
CREDIT LIMIT EXCEEDED			
1002	FRED	54	60
1003	MARK	68	65
CREDIT LIMIT EXCEEDED			
1004	JANE	10	40
1005	ANDY	18	20
1006	PAUL	98	50

CREDIT LIMIT EXCEEDED.

18/20

Andreas Virpuls 4PI

Documentation

The Program uses a 2-Dimensional array to produce a table to show deliveries of milk, eggs, cream, bread, and potatoes, by a milkman over a period of 1 week to one house.

Type of Program

It is a looping program containing 2 nested loops, and using a 2-Dimensional array to produce the table of results. All variables are stored as Strings.

Description

The Program produces a table (6 colums and 8 rows) using a 2-Dimensional array, and also the units of the substances in the table as a foot note. All variables including numbers, and the column headings (Milk, Eggs, cream, bread, and potatoes) and the row headings (Day, Mon, Tue, Wed, Thu, Fri, Sat, Sun) are stored as Strings (M$) to make it easier to print the table. Input is via Read and Data Statements. The table shows deliveries of milk eggs, cream, bread, and potatoes by a milkman in one week to one house..

Data preparation

All the data in the program is held in data Statements as Strings. While running the program the is no need to enter any numbers or names. All the data is in the range;

(i) 2 - 3 for pints of milk
(ii) 0 - 1 for dozens of eggs
(iii) 1 - 2 for loaves of bread
(iv) 0 - 2 for cartons of cream
(v) 0 - 10 for Pounds of potatoes

The data can be changed but it cannot be bigger than 9999 or have more than 4 digits (The decimal point counts as a digit) or every thing in that column will have to be changed to make up and make the table even.

Operating Instructions

This program should run on most machines, using Basic. It was written in BASIC for use ~~of~~ on the Research Machines Limited 380 Z and Link 480Z machine in 80 characters a line mode. To put the 480Z into 80 characters turn it on, or press the reset button if it is already turned on and press W then R. Type in the program, pressing the button marked "RETURN" after each line. When you have finished typing it in type "RUN" and press the RETURN button

Formulae

The program contains no calculations.

Results format:

The results are given in the form of a table with 6 columns and 8 rows. The headings are:-

across the top ↓

DAY	MILK	EGGS	CREAM	BREAD	POTATOES
MON					
TUE					
WED					
THU					
FRI					
SAT					
SUN					

← Down the side

The numbers in the columns represent the number of each product bought in one day.

Error Messages

NONE ~~except~~ if the program is correctly entered

Algorithm.

1 State how big the table is going to be.
2. Get the information
3 Print the information
4 Print Footnote

1st Refinement

1.1 State how many columns there are going to be.
1.2 State how many rows there are going to be
2 Input the information
3 Print the Information
4 Print foot note

2nd Refinement.

1.1 State how many columns there are going to be (6)
1.2 State how many rows there are going to be (8).
2.1 Input Headings
2.2 Input Name of day and the numbers
3.1 Print Headings
3.2 Print Name of day and the numbers
4 Print footnote showing units

3rd Refinement.

1.1 State how many columns there are going to be (6)
1.2 State how many rows there are going to be (8)
2.1 Input headings

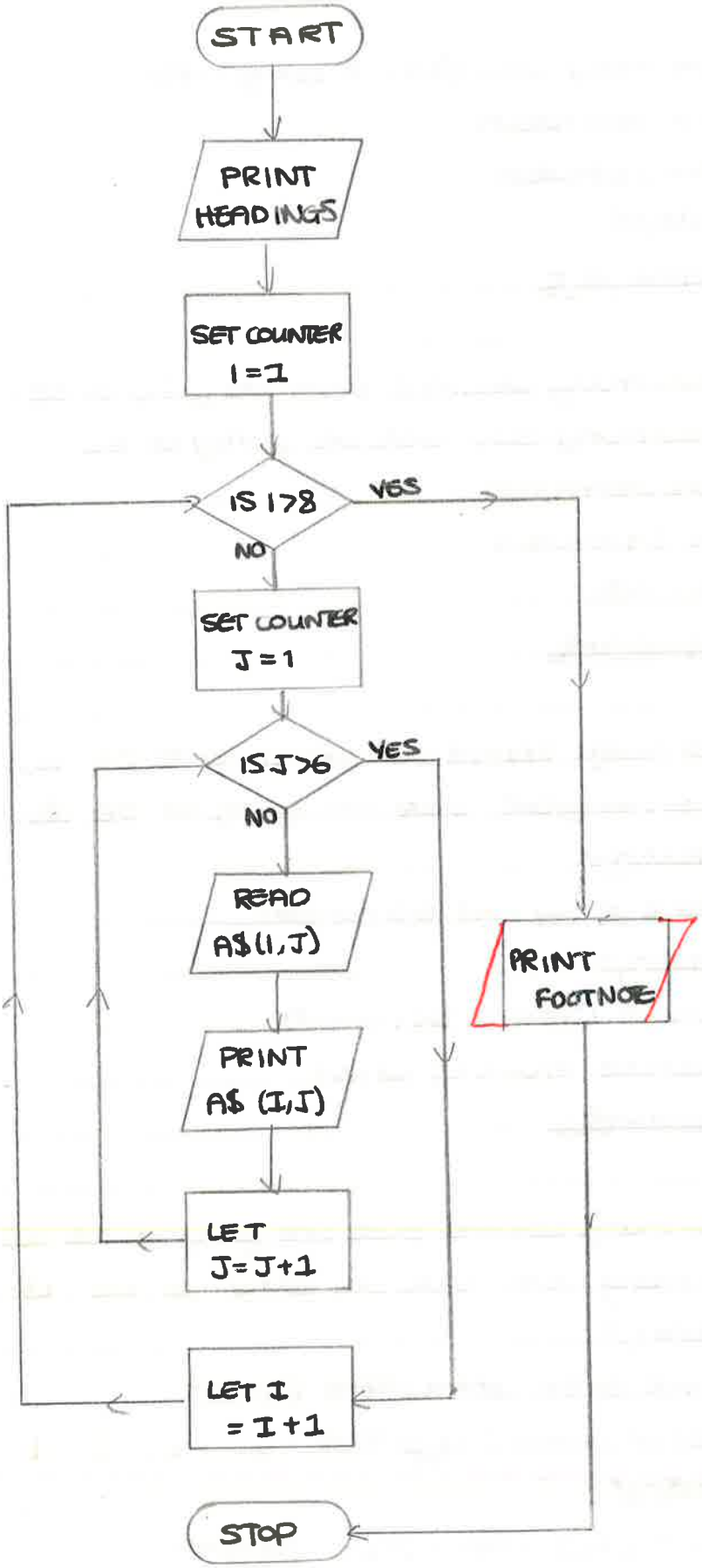
DAY MILK EGGS CREAM BREAD POTATOES

2.2 Input day and numbers (e.g MON 6 4 2 1 0)
3.1 Print Headings

DAY MILK EGGS CREAM BREAD POTATOES

3.2 Print day and numbers (e.g MON 6 4 2 1 0)
4 Print footnote showing units, (e.g milk = Pints etc).

Flowchart:


START
PRINT HEADINGS
SET COUNTER I=1
IS I>8
YES
NO
SET COUNTER J=1
IS J>6
YES
NO
READ A$(I,J)
PRINT A$(I,J)
LET J=J+1
LET I=I+1
PRINT FOOTNOTE
STOP

Program

Language : BASIC (Beginner Allpurpose Symbolic Instruction Code)

```
10  PUT 31
20  REM MILKMANS ROUND
30  DIM A$ (8,6)
40  FOR I = 7 TO 8
50  FOR J = 1 TO 6
60  READ A$ (I,J)
70  PRINT A$ (I,J);
80  NEXT J
90  PRINT
100 NEXT I                    105 PRINT
110 PRINT " MILK = PINTS "
120 PRINT " EGGS = HALF DOZENS "
130 PRINT " CREAM = CARTONS "
140 PRINT "BREAD = LOAVES "
150 PRINT "POTATOES = POUNDS "
160 DATA " DAY ", "MILK", " EGGS ", "CREAM", "BREAD", "POTATOES "
170 DATA " MON ", 2, Ø, Ø, 1, Ø
180 DATA "TUE ", 3, Ø, Ø, 1, 4
190 DATA " WED ", 2, Ø, 1, 1, Ø
200 DATA " THU ", 3, Ø, Ø, 1, Ø
210 DATA " FRI ", 2, 2, Ø, 1, Ø
220 DATA " SAT ", 3, Ø, 1, 2, 4
230 DATA "SUN ", 2, Ø, 1, Ø, Ø
240 END
```

6

Expected Results

DAY	MILK	EGGS	CREAM	BREAD	POTATOES
MON	2	0	0	1	0
TUE	3	0	0	1	4
WED	2	0	1	1	0
THU	3	0	0	1	0
FRI	2	2	0	1	0
SAT	3	0	1	2	4
SUN	2	0	1	0	0

MILK = PINTS

EGGS = HALF DOZENS

CREAM = CARTONS

BREAD = LOAVES

POTATOES = POUNDS

Actual Results

✓

3