

SECTION 4

Theory and techniques of Programming

Beginners
All Purpose
Symbolic
Instruction
Code

Statements

1 REM (remark)

10 REM

2 PRINT

10 PRINT "Name" = Name

10 PRINT 2+2 = 4

10 PRINT "2+2" = 2+2

10 ? "Name" = Name

3 LET

10 LET A = B+C

4 END

To finish program

SAMPLE PROGRAM

10 REM THIS IS A SAMPLE PROGRAM

20 LET A = 9/3

30 PRINT A = 3

40 END

5 INPUT

10 INPUT B = ? 4

20 LET A = 3*B = 3x4

30 PRINT A = 12
 40 END

e.g.

10 INPUT B,C = 2,3

10 INPUT B = ? A

20 INPUT C = ? B

6 READ, DATA

10 REM READ, DATA

20 READ A,B,C

30 LET D = A+B+C

40 PRINT D = 9

50 DATA 2,3,4

60 END

7 FOR....., NEXT

10 REM LOOPING PROGRAM

20 FOR I = 1 TO 5 (STEP 1)

30 READ A

40 LET B = A/6

50 PRINT B = 1,2,3,4,5 (on different lines)

60 NEXT I

70 DATA 6,12,18,24,30

80 END

Arithmetic Functions

<u>Operation</u>	<u>Algebra</u>	<u>BASIC</u>
Brackets	()	()
Powers + roots	$x^2 \sqrt{x}$	$x \uparrow 2 \quad x \uparrow (1/2)$
Multiplication	x	*
Division	÷	/

Addition	+	+
Subtraction	-	-

In order of priority, top first

Problem

Each working day at a factory a record is made of the production of machines. For each machine a data card is produced with the date, machine number, number of items produced, and number of running hours. Write a program to read 8 such data cards and calculate the number of items produced per hour (the production rate for each machine

PRINT table with headings $R = \frac{P}{I} = R = \text{prod rate}, P = \text{no of items}$
I

T = hours running. The dates = D\$ = "22/9/83"

```

10 REM DATA CARDS
20 PRINT ? "DATE**** PRODUCTION **** ITEMS * TIME ** MACHINE"
30 ? "***** * RATE***** ** MADE ** TAKE N * NUMBER*"
40 FOR N = 1 TO 8
50 LET T = 8
60 LET D$ = "22/9/83"
70 READ P, M
80 LET R = P / T
90 ? D$; "*" ; R; "I/H**** *"; P; "*" ; T; "HS***"; M; "**"
100 NEXT N
110 DATA 8000, 1
120 DATA 8064, 2
130 DATA 8016, 3
140 DATA 8056, 4
150 DATA 8024, 5
160 DATA 8008, 6
170 DATA 8032, 7
180 DATA 8040, 8
190 END

```

should be outside loop

loop

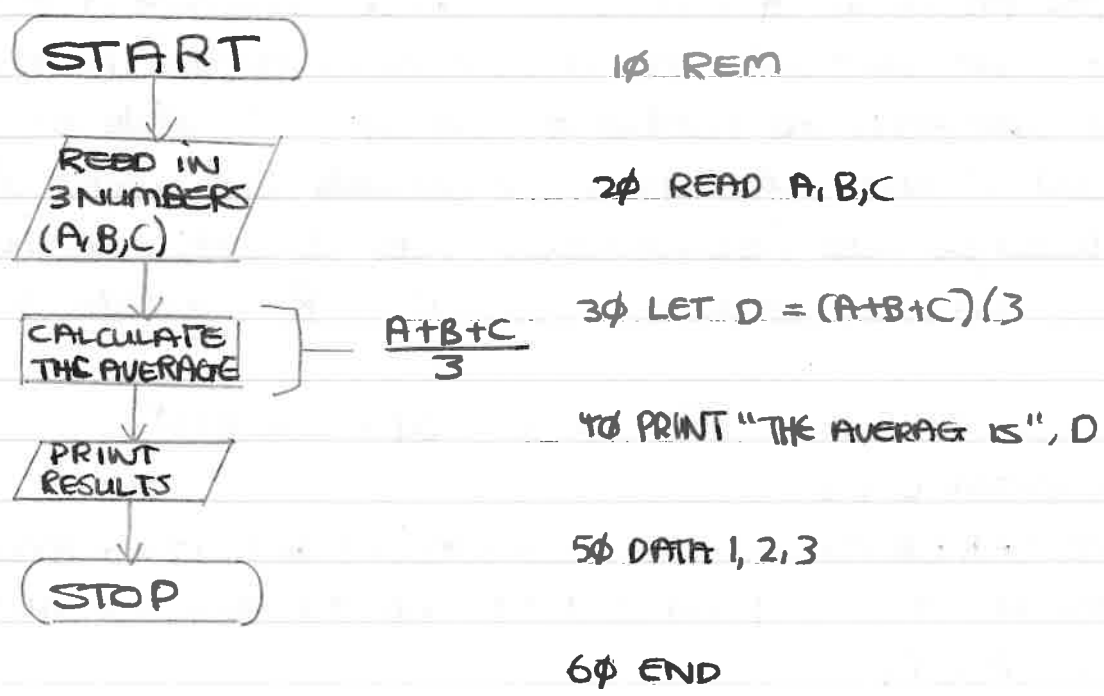
?

?

(B)

162B N°5
~~R = Rate Per Hour
P = Pay
H = Hours
N\$ = Name
W = Worker Number~~

Linear Program



Homework Problem

R = Rate of Pay

P = Pay

W = Worker Number

H = Hours worked

N\$ = Name of Worker

$$P = R * H$$

10 REM PAYROLL

20 ~~?~~ "NAME**RATE****PAY(E)*HOURS**WORKS*"

30 ~~?~~ "*****PER* HOUR * ***** WORKED * NUMBER"

40 FOR N = 1 TO 10

50 LET H = 40

60 READ N\$, R, W

70 LET P = R * H

80 ~~?~~ ^{PRINT} N\$, " ", R, "****", P, "****", H, "****", W, "*" ✓

90 NEXT N.

100 DATA "BILL", 1.25, 0.1

110 DATA "JOHN", 2.15, 0.2

120 DATA "FRED", 1.15, 0.3

130 DATA "MARK", 1.75, 0.4

140 DATA "ANDY", 2.25, 0.5

150 DATA "BERT", 2.05, 0.6

160 DATA "DICK", 1.55, 0.7

170 DATA "DEAN", 1.75, 0.8

180 DATA "JACK", 1.05, 0.9

190 DATA "PAUL", 1.55, 1.0

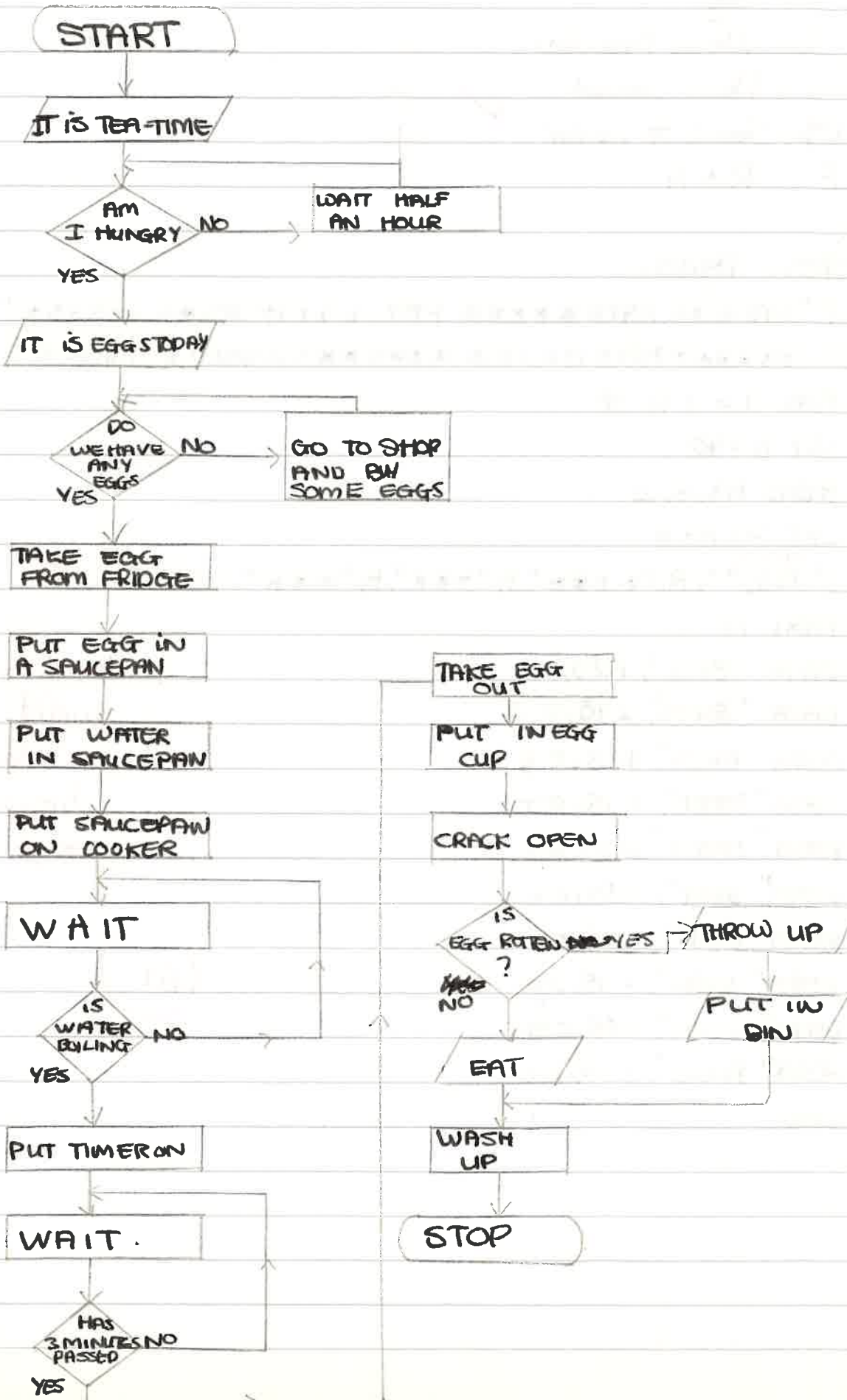
200 END

However they might have worked different hours, eg 4 ill, or part time etc.

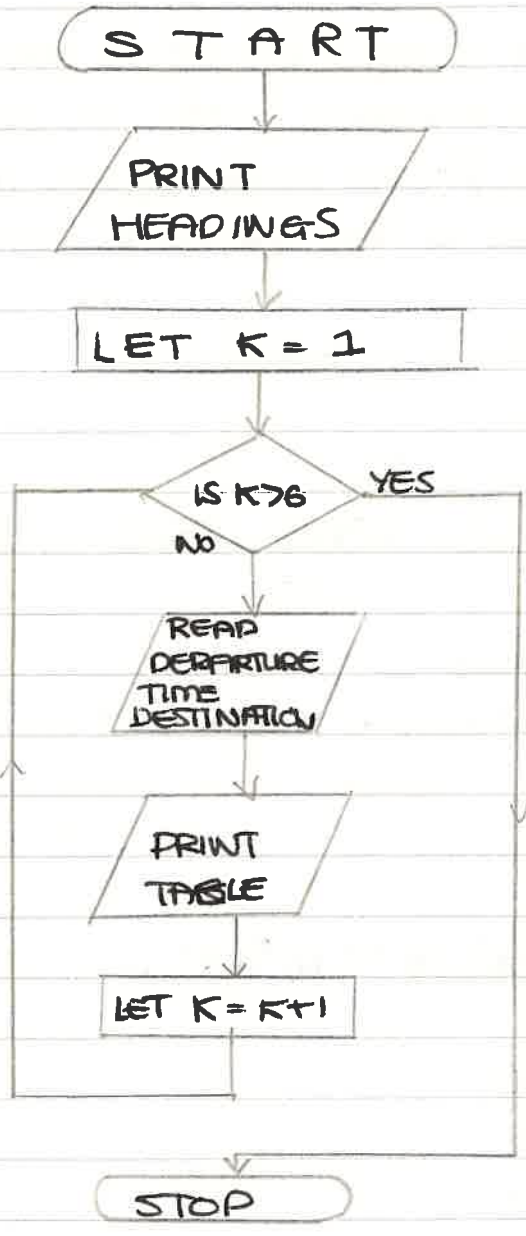
BT

Flow-Charting

Boil a three minute egg:



Flowchart



Homework

A coal merchant charges £9.54 per ton of coal. For each order a data card is prepared with the order number (R) Customers name (C\$) and number of tons (N)

Write a program to read the cards for 5 orders and calculate the cost

$$C = 9.54 \times N \quad * = \text{Space}$$

$$C = \text{cost } (£)$$

Heading = ORDER NO. NAME TONS COST (£)

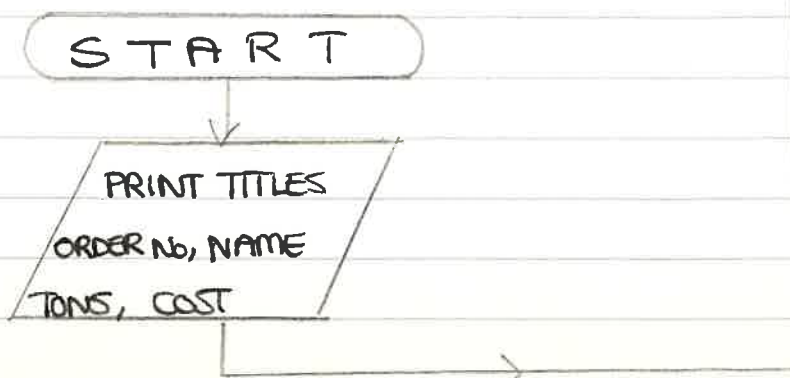
```

10 REM COAL DELIVERIES
20 PRINT "ORDER*No** NAME* TONS* COST (£)"
30 FOR (N) = 1 TO 5
40 READ R, C$, (N)
50 LET C = 9.54 * N
60 PRINT "**", R, "****", C$, "", N, "**", C
70 NEXT N.
80 DATA 1, "BROWN", 1.5
90 DATA 2, "GREEN", 1.0
100 DATA 3, "WHITE", 0.5
110 DATA 4, "SMITH", 2.0
120 DATA 5, "YOUNG", 2.5
130 END
    
```

cannot have these variables the same

* = Space.

Flow chart for the above program



~~WORK OUT COST~~
~~FOR 5 ORDERS~~
LET N = 1
(FOR N = 1 TO 5)

IS N ⁷⁵ GREATER THAN 5

READ IN THE
ORDER NO, NAME
OF CUSTOMER, TONS

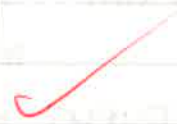
CALCULATE COST (C)
 $C = 9.54 \times N$

PRINT
RESULTS

LET N =
N + 1

STOP

BT



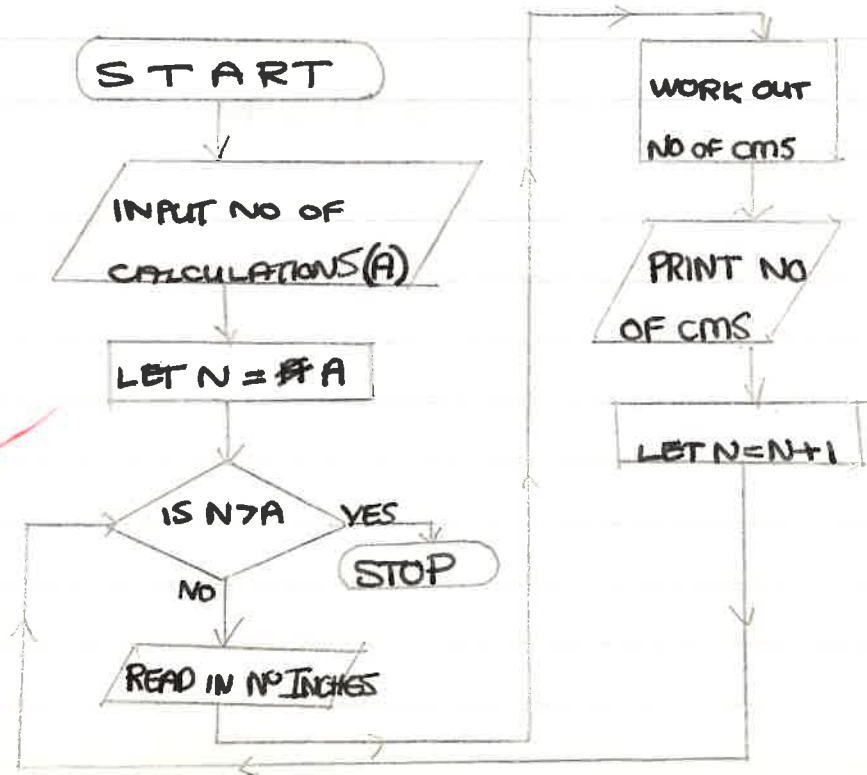
Problem

Write a program + flowchart to convert values input in inches to centimetres and print the results (1 inch = 2.54cm)

```
10 REM CONVERSION
20 PRINT "THIS PROGRAM CONVERTS IMPERIAL INCHES TO METRIC
    CENTIMETRES"
30 PRINT "HOW MANY CALCULATIONS DO YOU WANT TO
    MAKE"
40 INPUT A
50 FOR N = 1 TO A (STEP 1)
60 PRINT "ENTER THE NUMBER OF INCHES"
70 INPUT B
80 PRINT "THANK YOU"
90 LET C = B * 2.54
100 PRINT B " INCHES IS EQUAL TO " C " CENTIMETRES"
110 PAUSE 100
120 CLS
130 NEXT N
140 PRINT "THAT IS THE END"
150 END.
```

good - interesting additions - do they work!

good effort



IF THEN statement

Relations used in statements

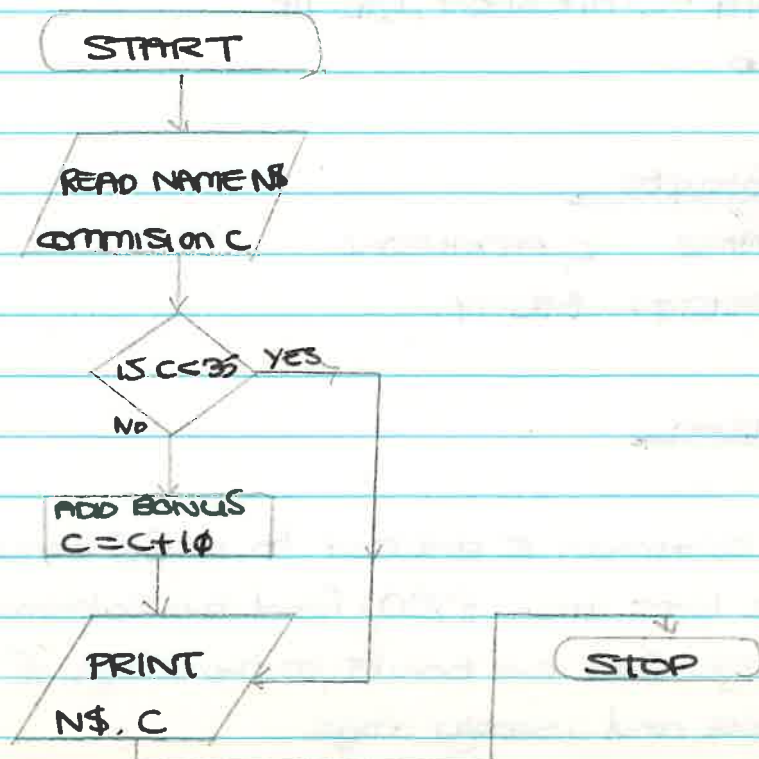
A condition statement also contains a comparison between two quantities. If the comparison is true, control passes to the statement indicated after the word THEN. If the comparison is not true, control passes to the next statement.

<u>Symbol</u>	<u>Meaning</u>
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
=	Equal to
<> or ≠	Not equal to

Example 4A1

A Salesman is entitled to a bonus of £10 unless his commission for a weeks sales is less than £35. Input a Salesman's name and commission, and add the bonus if necessary

Flow chart



Method

The condition is:

is the commission less than £35?

If so bonus is added.

Variables

N\$: Name, C: commission

Program

```
1  REM BASIC EXAMPLE 4A1
5  READ N$, C
10 IF C < 35 THEN 20
15 LET C = C + 10
20 PRINT "NAME"; N$
25 PRINT "COMMISSION"; C
30 STOP
35 DATA "C. ATKINSON", "£46.74"
40 END
```

Results

NAME C. ATKINSON
COMMISSION £56.74.

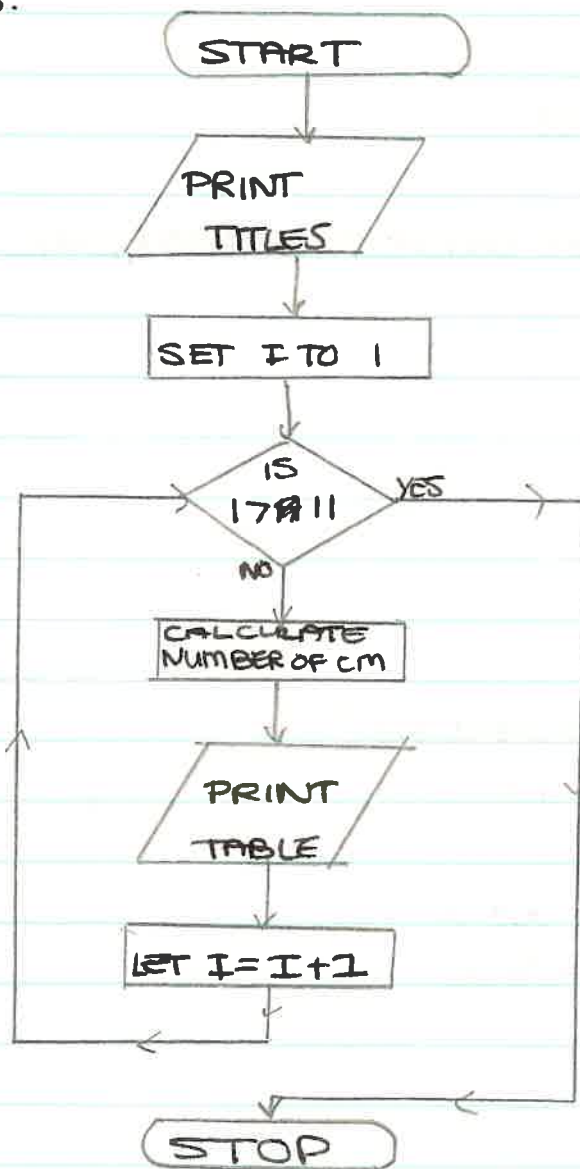
Problem

A salesman is entitled to a £15 bonus unless his weekly sales are less than £200. Read the salesman's name, sales and weekly wage. Add the bonus to the wage if necessary. Print the name and weekly wage.

Problem

Write a program + flowchart to print a conversion table

```
10 REM CONVERSION TABLE
20 ? " INCHES * CENTIMETRES "
30 FOR I = 1 TO 12
40 LET C = I * 2.54
50 ? I; "*****"; C
60 NEXT I
70 END.
```



Problem

Write a program + flowchart to calculate the wages of 10 ten people given the following information Hours Worked and Rate per hour

Find out the current conversion rates for Pounds Sterling to a number of foreign currencies such as Dollars, Marks, Francs, Lira. Input the names of say 4 foreign currencies and their conversion rates. Print a table of conversion from sterling to the foreign currencies over the range £1-50.

```
10 REM EXCHANGE RATES.
20 ? "EXCHANGE RATES 3.11.83."
30 ?
40 ? "STERLING ** PESETA ** DRACHMA ** YUGO * DINAR."
50 FOR S = 1 TO 50
60 LET P = S * 226
70 LET D = S * 147
80 LET Y = S * 204 ✓
90 ? LET A$ = "*****"
100 LET B$ = "***"
110 LET C$ = "****"
120 IF S > 9 THEN LET A$ = "*****"
130 IF P > 999 THEN LET B$ = "***"
140 IF P > 9999 THEN LET B$ = "*"
150 IF D > 999 THEN LET C$ = "****"
160 IF D > 9999 THEN LET C$ = "**"
170 ? S; A$; P; B$; D; C$; Y ✓
180 NEXT S
190 END.
```

complicated way of producing a table!

* = Space.

When screen gets full press spacebar for more information.

Variable

N = counter (1 to 6) S: Stack Number

C = Moisture content

Programme

```
5 ? " STACK NO, ", " MOISTURE CONTENT "  
10 FOR N = 1 TO 6 (STEP 1)  
15 READ S, C  
20 IF C < 15 THEN 35  
25 ? S, C  
30 GOTO 40  
35 ? S, C, "READY FOR CUTTING "  
40 NEXT N  
50 DATA 475, 13  
55 DATA 613, 17  
60 DATA 74, 11  
65 DATA 129, 8  
70 DATA 518, 23  
75 DATA 117, 15  
80 END
```

Results

STACK NO	MOISTURE CONTENT	
475	13	READY FOR CUTTING
613	17	
74	11	READY FOR CUTTING
129	8	READY FOR CUTTING
518	23	
117	15	

Points to notice

The IF and GOTO statements are both inside the loop.

Conditions in Loops

Example 4BI

At a timber yard, wood is dried until its moisture content is below 15%. If so it is ready to be cut. A data card is prepared for each of six samples, showing its stack number and moisture content. Read and print this information, with the message READY FOR CUTTING when required.

Method

Headings are printed:

STACK NO. MOISTURE CONTENT (%)

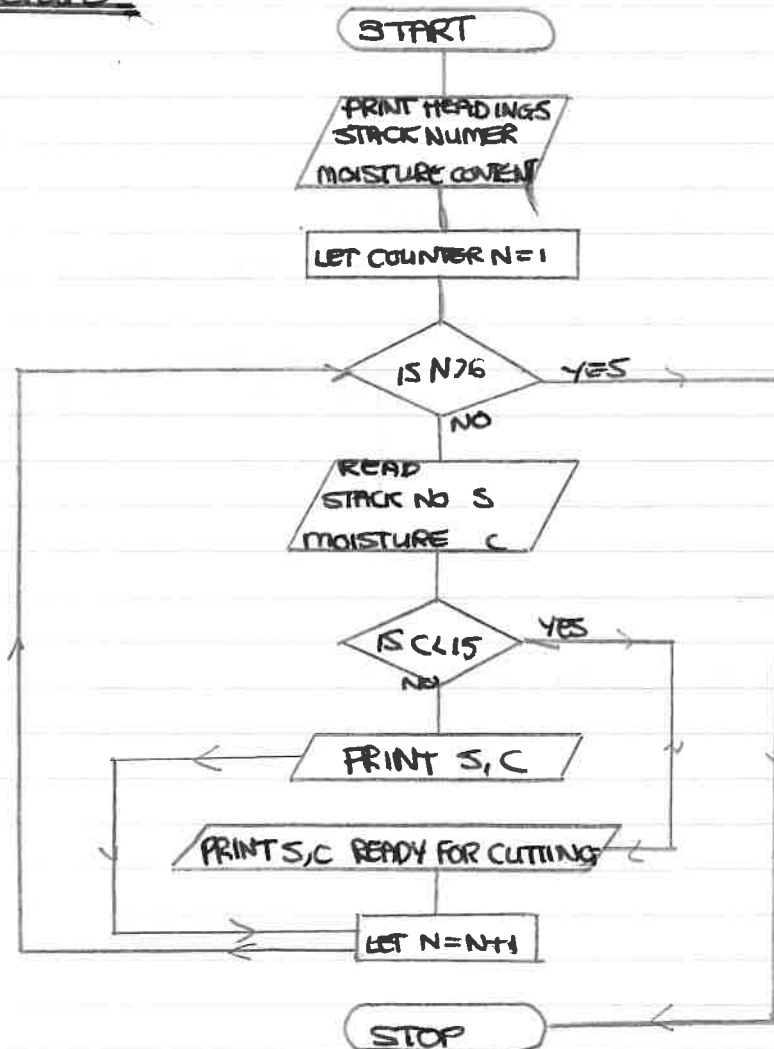
A loop counter from 1 to 6) is used to work through the data

The condition

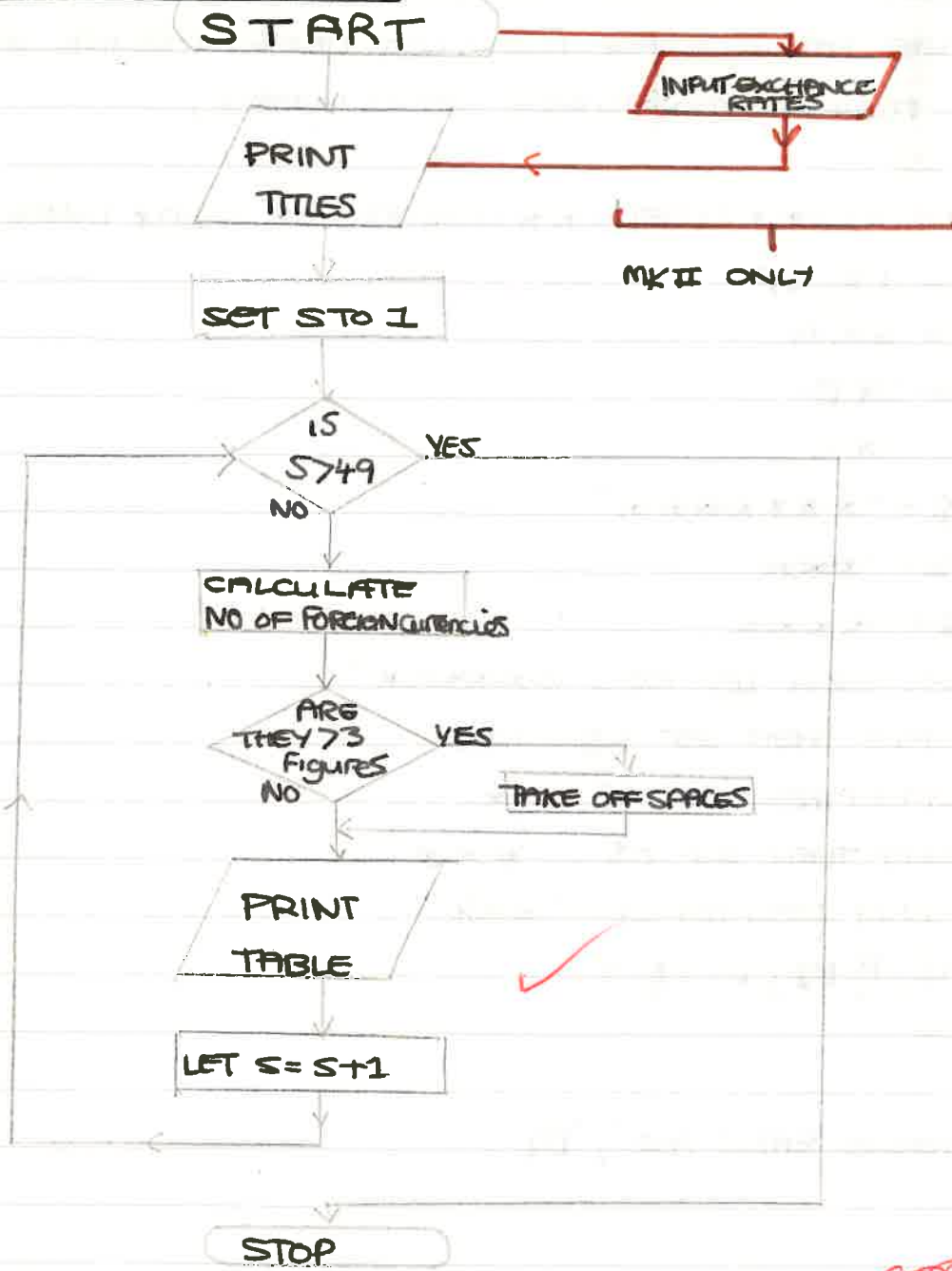
is the percentage less than 15%

sorts the stacks which are ready to be cut. The data is printed with the message if required

Flowchart



Andrew Virtuous 4PI



good (A-)

Flow chart for exchange rate program

- 1φ REM EXCHANGE RATES MK II
- 2φ ? "WHAT IS THE DATE?"
- 3φ INPUT D\$
- 4φ ? "WHAT IS THE EXCHANGE RATE FOR THE PESETA? (ABOUT 220 - 230 - ROUND IT UP TO THE NEAREST WHOLE NUMBER)"
- 5φ INPUT A
- 6φ ? "WHAT IS THE RATE FOR THE DRACHMA?" (AROUND 250 - ROUND IT UP TO THE NEAREST WHOLE NUMBER)"

```

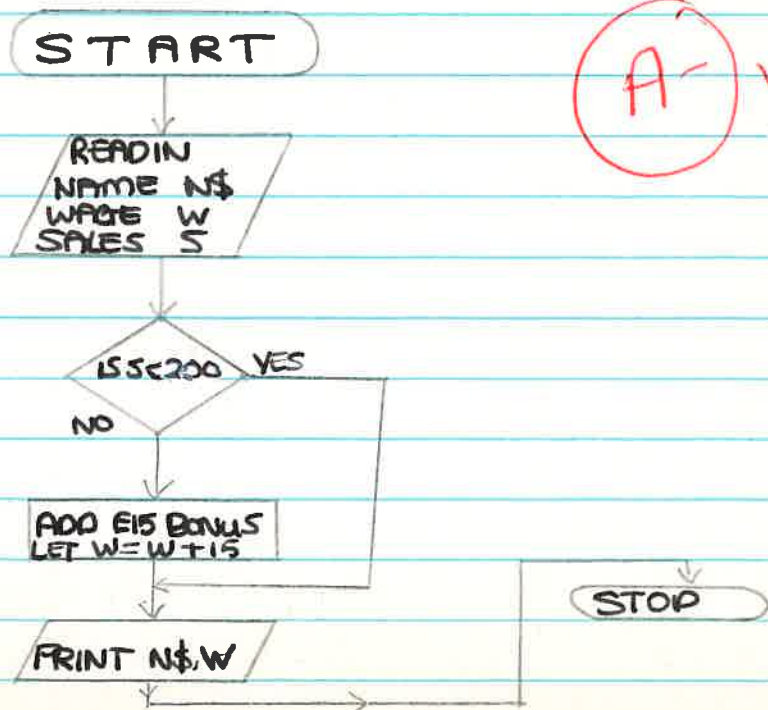
70 INPUT B
80 ? "ENTER THE RATE FOR THE YUGO DINAR? (AROUND 200 -
    ROUND IT UP TO THE NEAREST WHOLE NUMBER)
90 INPUT C
100 ? "STERLING** PESETA** DRACHMA** YUGO* DINAR"
110 FOR S = 1 TO 50
120 LET P = S * A
130 LET D = S * B.
140 LET Y = S * C
150 LET A$ = "*****"
160 LET B$ = "***"
170 LET C$ = "****"
180 IF S > 9 THEN LET A$ = "*****"
190 IF P > 999 THEN LET B$ = "**"
200 IF P > 9999 THEN LET B$ = "*"
210 IF D > 999 THEN LET C$ = "***"
220 IF D > 9999 THEN LET C$ = "**"
230 ? S, A$, P, B$, D, C$, Y
240 NEXT S
250 END
95 ? "EXCHANGE RATES FOR"; D$
97 ?

```

* = Space.

Program

```
10 REM BONUS
20 ? "ENTER YOUR NAME"
30 INPUT N$
40 ? "WHAT IS YOUR WEEKLY WAGE"
50 INPUT W
60 ? "WHAT IS YOUR SALES TOTAL"
70 INPUT S
80 ? "THANK YOU"
90 FOR A = 1 TO 200: NEXT A
100 PUT 12
110 IF S < 200 THEN 140
120 LET W = W + 15
130 ? "YOU HAVE EARNED A BONUS"
140 ?
150 ? "SALESMAN'S NAME"; N$
160 ?
170 ? "TOTAL WAGE "; "E"; W
180 ?
190 ? "IF YOUR SALES WERE MORE THAN £200 YOU WILL HAVE
    GAINED A £15 BONUS."
200 END
```



A- very good

Project

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

Project Name

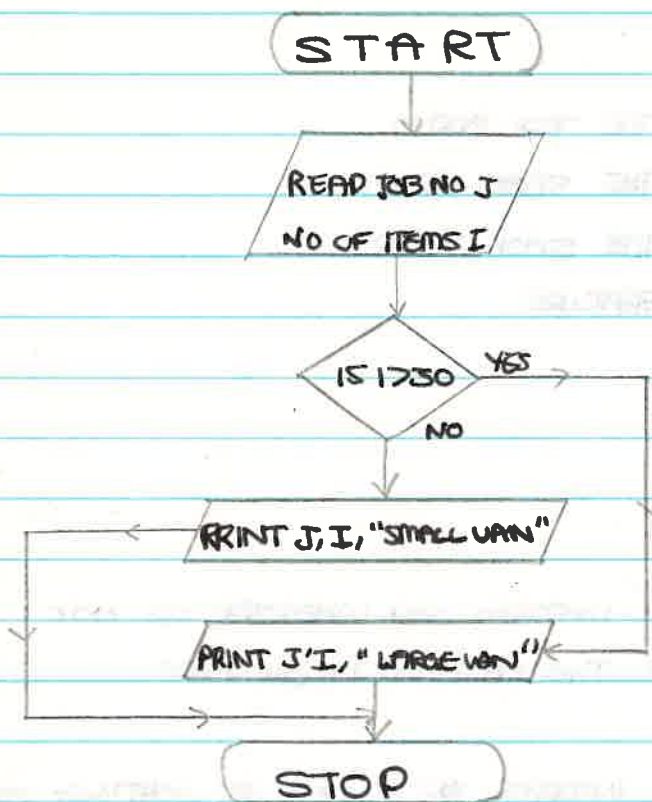
Project Name

Conditions

Problem

A removal firm has two sizes of vans: small and large. Small vans for up to 30 items, large vans for over 30 items. Read a job number and a number of items. Print this information together with the message LARGE VAN or SMALL VAN as required

Flow-chart



Program

```
10 REM FURNITURE REMOVAL
20 READ J,I
30 IF I > 30 THEN 60
40 PRINT J,I, "SMALL VAN."
50 GOTO 70
60 PRINT J,I, "LARGE VAN"
70 DATA 4617, 46
80 END
```


Problem

The heating system of a factory is switched on if the average temperature at 3 points is less than 13°C . Read three temperatures, calculate the average, print the average and the message HEATING ON or HEATING OFF as required

Condition = $T > 13$

Variables

A = TEMPERATURE IN THE TOOL ROOM

B = TEMPERATURE IN THE STORE ROOM

C = TEMPERATURE ON THE SHOP FLOOR

T = AVERAGE TEMPERATURE

Conditions Used

120 IF $C > 40$ THEN 240

130 IF $B > 40$ THEN 240

140 IF $A > 40$ THEN 240

} INFORM YOU WHETHER OR NOT
THE FACTORY IS ON FIRE

180 IF $T > 13$ THEN 200 - INFORMS YOU IF THE HEATING HAS
BEEN TURNED ON *

190 IF $T < 13$ THEN 270 - INFORMS YOU IF THE HEATING HAS
BEEN TURNED OFF *

* THE HEATING IS TURNED ON IF THE AVERAGE TEMPERATURE
IS BELOW 13°C

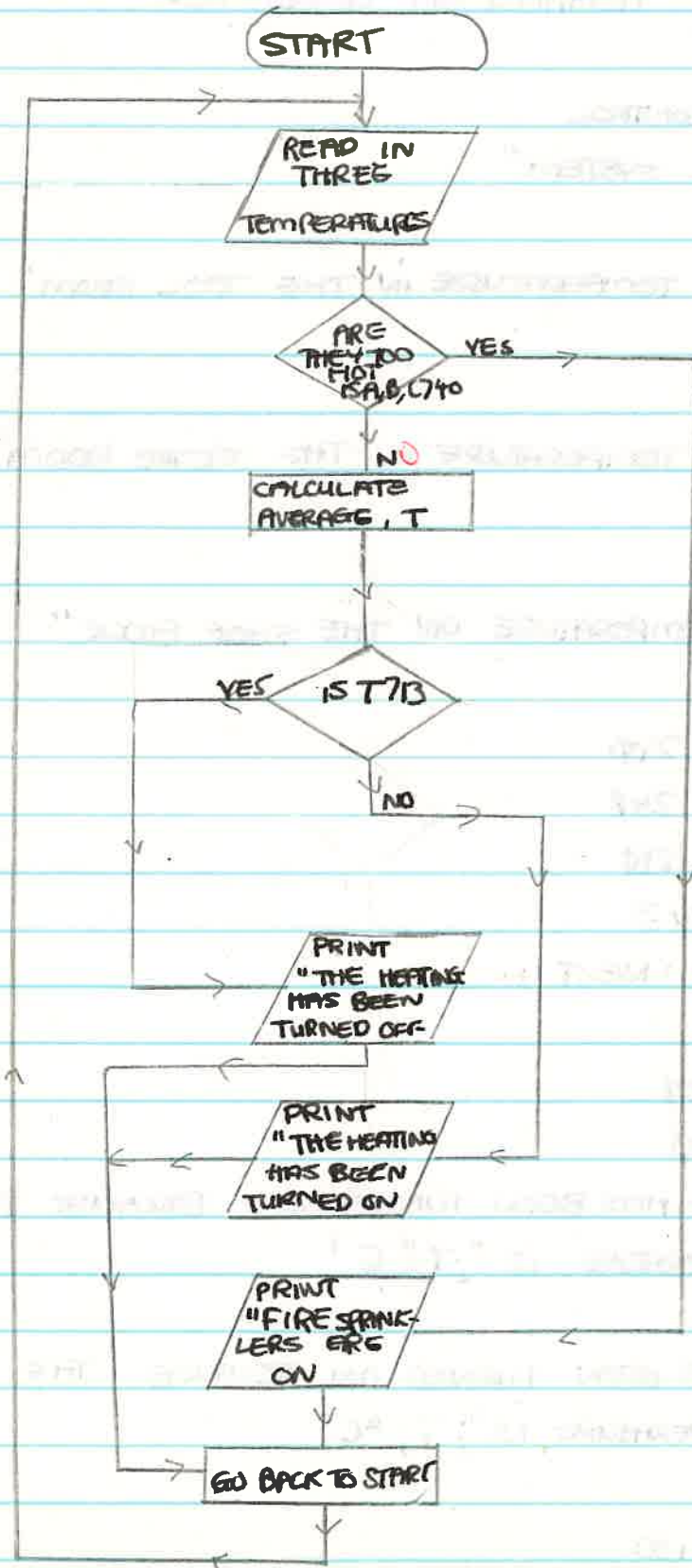
THE HEATING IS TURNED OFF IF THE AVERAGE TEMPERATURE
IS ABOVE 13°C

Problem

The heating system of a factory is switched on if the average temperature at 3 points is less than 13°C . Read three temperatures calculate the average, print the average and the message HEATING ON or HEATING OFF as required.

```
10 REM HEATING CONTROL
20 ? "BOILER CONTROL SYSTEM"
30 ? : ?
40 ? "WHAT IS THE TEMPERATURE IN THE TOOL ROOM"
50 INPUT A
60 ?
70 ? "WHAT IS THE TEMPERATURE IN THE STORE ROOM"
80 INPUT B
90 ?
100 ? "WHAT IS THE TEMPERATURE ON THE SHOP FLOOR"
110 INPUT C
120 IF C > 40 THEN 240
130 IF B > 40 THEN 240
140 IF A > 40 THEN 240
150 LET T = (A+B+C)/3
160 FOR N = 1 TO 150 : NEXT N
170 PUT 31
180 IF T > 13 THEN 200
190 IF T < 13 THEN 220
200 ? " THE HEATING HAS BEEN TURNED OFF BECAUSE THE
    AVERAGE TEMPERATURE IS "; T; " C "
210 GOTO 220
220 ? " THE HEATING HAS BEEN TURNED ON BECAUSE THE
    AVERAGE TEMPERATURE IS "; T; " C "
230 GOTO 290
240 FOR F = 1 TO 150
250 ? " FIRE ";
```


260 NEXT F
 270 ? "THE SPRINKLERS ARE ON"
 280 FOR J = 1 TO 200 : NEXT
 290 PUT 31
 300 GOTO 10



Very good!
 (A)

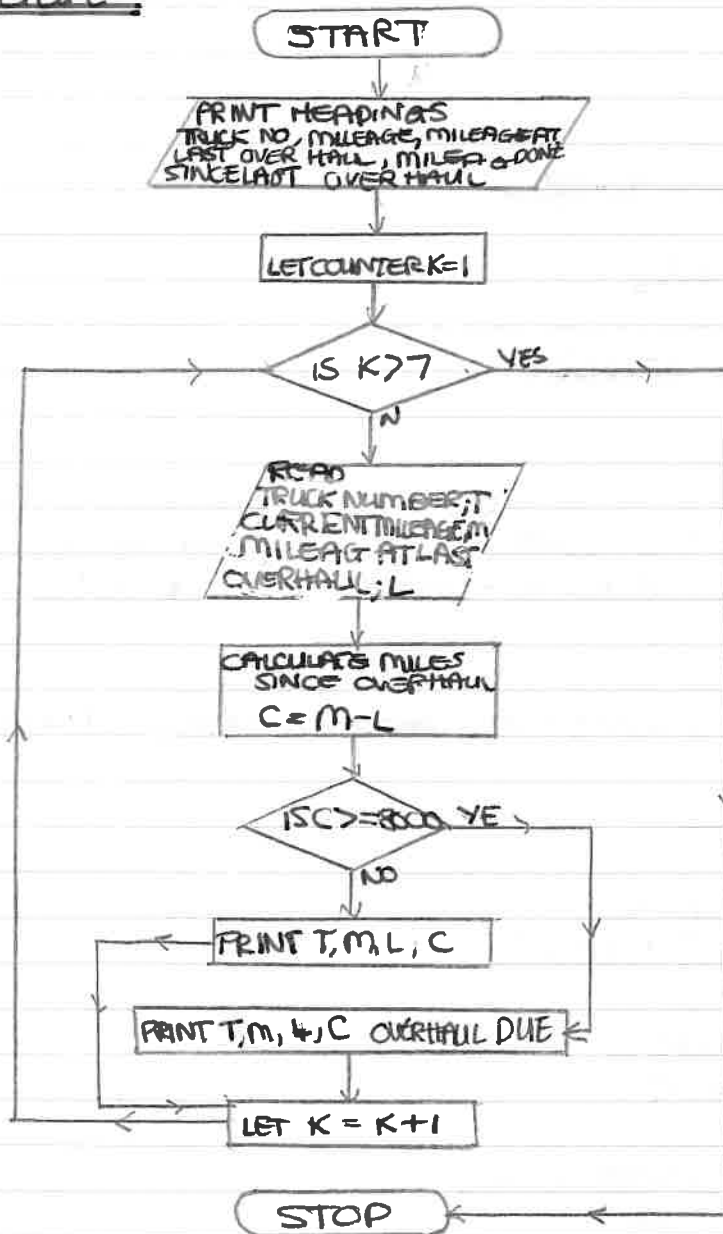
Exercise 4B question 1

A transport firm overhauls its trucks every 8000 miles. For each of seven trucks (variable T), current mileage, (variable M) and mileage at last overhaul (variable L) are read

The number of miles travelled since the last overhaul, (Variable C , $C = M - L$) is calculated. If this is equal to or greater than 8000 miles an overhaul is due.

The variables T, M, L , and C are printed with the message "overhaul due" if necessary

Flowchart



5 ? "TRUCK NO.", "CURRENT MILEAGE", "MILEAGE AT LAST OVERHAUL", "MILES TRAVELLED SINCE LAST OVERHAUL"

10 FOR K = 1 TO 7 STEP 1

15 READ T, M, L

20 LET C = M - L

```

25 IF C>8000 THEN 40
30 ? T, M, L, C
35 GOTO 45
40 ? T, M, L, C, " OVERHAUL DUE "
45 NEXT K
50 STOP
55 DATA 7, 16715
60 DATA 5, 69412
65 DATA 1,
70 DATA 6,
75 DATA 4,
80 DATA 2,
85 DATA 3,
90 END

```

Programming

DIM statement

i) LISTS (1 D array)

Problem

Read lists of the reference numbers and names of six items kept at a shop. Print a table of this information

Method

A list R(6) stores the reference numbers, N\$(6) the names

Program

```

10 DIM R(6), N$(6)
20 PRINT "REF NO", "ITEM"
30 FOR K = 1 TO 6
40 READ R(K), N$(K)
50 PRINT R(K), N$(K)
60 NEXT K
70 DATA 4167, "POWER DRILL"
80 DATA 5215, "SAW"
90 DATA 6891, "HAMMER"
100 DATA 2935, "CHISEL"
110 DATA 3428, "PLANE"
120 DATA 1984, "PLIERS"
130 END

```

Flowchart

START

SET UP DIM
STATEMENT
R(K) N\$(0)

PRINT
HEADINGS
REFNO
ITEM

SET K = 1

IS K > 6
YES
NO

READ
Ref No R
Name N\$

PRINT
Ref No R
Name N\$

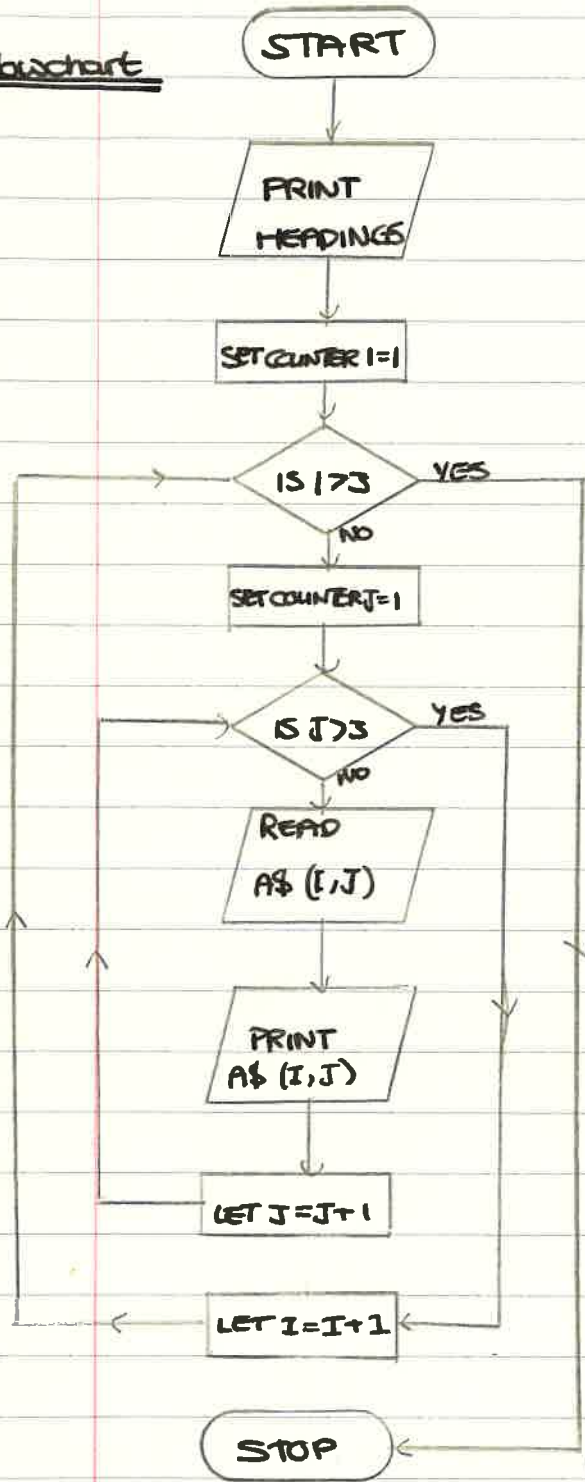
LET K = K + 1

STOP



2D Array

Flowchart



10 DIM A\$(3,3)

20 ? "SALES FIGURES"

30 FOR I = 1 TO 3 STEP 1

40 FOR J = 1 TO 3

50 READ A\$(I,J)

60 ? A\$(I,J),

70 NEXT J

80 PRINT

90 NEXT I

100 DATA "BEER", "1982", "1983"

110 DATA "BOTTLED", "10", "20"

120 DATA "DRAUGHT", "30", "50"

130 END

Results

Subs Figures

Beer	1982	1983
Bottled	10	20
Draught	30	50

Spacing

In above program enter :-

```
60 ? TAB (P) A$(1,J);
65 LET P = P + 8
85 LET P = 0
```

Problem

Using a 2D array print a table to show deliveries of milk, eggs, cream, bread, and potatoes, by a milkman over a period of 1 week to 1 house.

Random Number Generation

```
10 REM RANDOM NUMBER GENERATION
20 RANDOMIZE
30 LET A = INT (10 * RND (1)) + 1.
40 PRINT A
50 END
```

RND (1) = 0-1 (including decimals)
10 * RND (1) = 0-9.99 (including decimals)
INT = 0-9
+ 1 = 0-10.

Example 2

```
10 REM RANDOM NUMBERS
20 ? "TYPE IN A POSITIVE NUMBER TO START GENERATOR"
30 INPUT J.
40 LET I = RND (-J)
50 ? "RANDOM NUMBER, RANGE 1-10"
60 LET X = INT (10 * RND (1)) + 1
70 ? X
80 END
```

The Numbers chosen by the computer are not entirely random. A positive argument will produce a sequence of numbers, that sequence depending on the positive number used. A negative argument will always produce the same number on the same computer. This fact is used to choose the positive random sequence.

Handwritten notes, possibly a list or table, with some illegible text and a large bracket on the right side.

Handwritten notes, possibly a list or table, with some illegible text.

Handwritten notes, possibly a list or table, with some illegible text and a large bracket on the right side.

Handwritten notes, possibly a list or table, with some illegible text.

Problem

19/20

Very good - interesting ideas

Each move in a game for two players is as follows. Each player throws a dice and the one with the higher one scores the point. There is no score if all the numbers are equal. Simulate ten moves of this game, adding up the total score for one player. At the start each player enters his lucky number to start the generator. The statement for one throw is

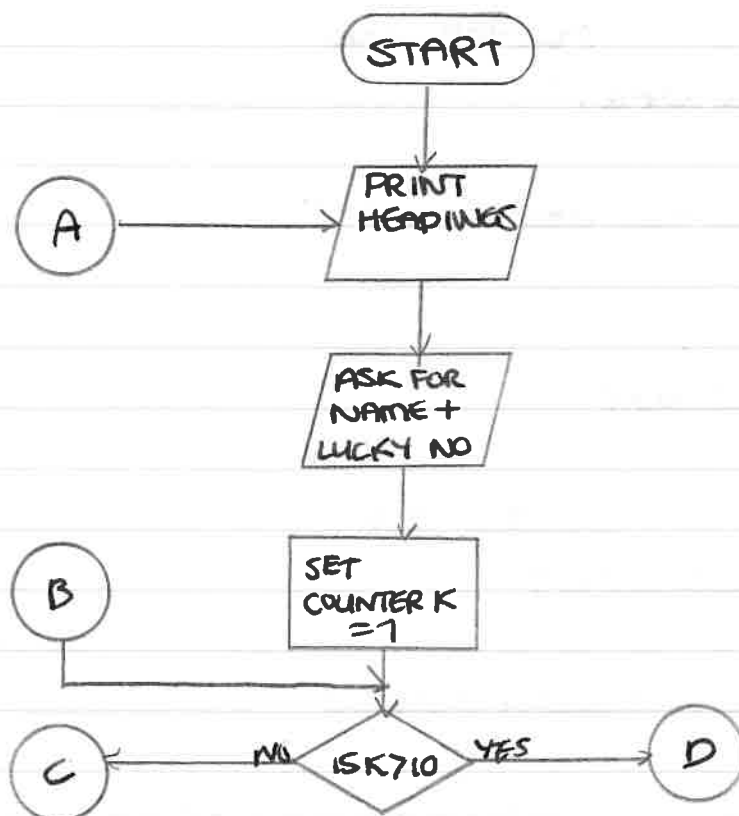
```
LET X = INT (6 * RND (1)) + 1
```

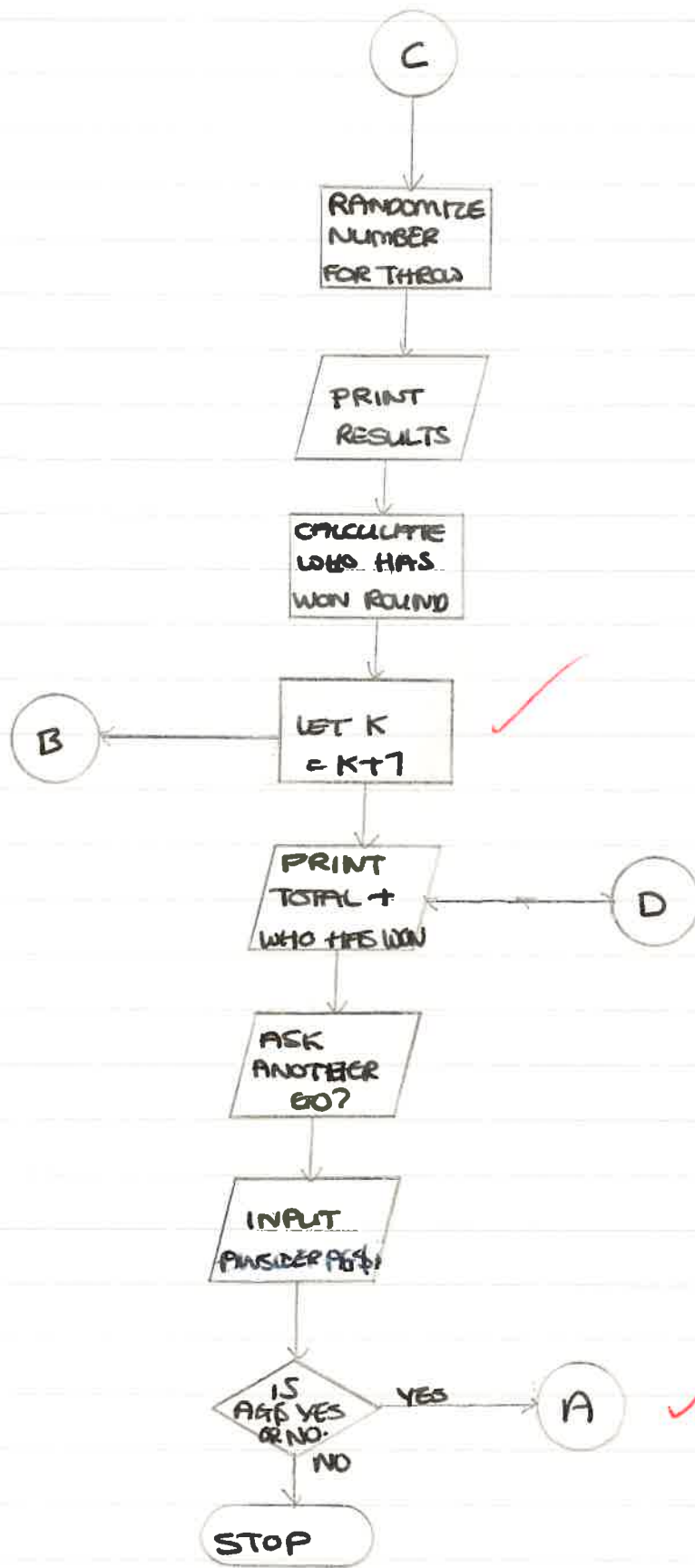
```
10 ? " - - - - - "
20 ? " | | "
30 ? " | * * | TEN THROW "
40 ? " | | DICE GAME "
50 ? " | | "
60 ? " | * * | ANDREW VIRNULS "
70 ? " | | "
80 ? " - - - - - "
90 LET A = 0
100 LET B = 0
110 ? : ? : ? " PLAYER ONE "
120 ? : ? " WHAT IS YOUR NAME ? "
130 INPUT N$
140 ? : ? " N$ : WHAT IS YOUR LUCKY NUMBER ? "
150 INPUT L
160 LET S = RND (-L)
170 FOR N = 1 TO 500 : NEXT
180 PUT 31
190 ? : ? : ? " PLAYER TWO "
200 ? : ? " WHAT IS YOUR NAME ? "
210 INPUT NA$
220 ? : ? NA$ " WHAT IS YOUR LUCKY NUMBER ? "
230 INPUT M
240 LET T = RND (-M) 245 ? : ? N$ : M$
```

```

280 FOR K = 1 TO 10
290 LET X = RND * INT (6 * RND (1)) + 1
300 LET A = A + X
310 LET Y = INT (6 * RND (1)) + 1
320 LET B = B + Y
330 ? A, B X, Y
340 IF X > Y THEN ? = N$; "WON THAT ROUND"
350 IF Y > X THEN ? = NA$; "WON THAT ROUND."
360 NEXT K
370 ? : ? : ? "TOTAL"
380 ? : ? A, B.
390 IF A > B THEN ? : ? = N$ "WON THAT GAME"
400 IF B > A THEN ? : ? = NA$ "WON THAT GAME"
410 IF A = B THEN ? : ? "THE GAME IS A DRAW"
420 ? : ? "ANOTHER GAME? (Y/N) "
430 INPUT AG$
440 IF AG$ = "Y" THEN ? @ PUT 31: GOTO 10
450 ? : ? "GOODBYE!"
460 END.

```





good construction


```

5 PUT 31
10 ?" _____ "
20 ?" |
30 ?" | * * | TEN THROW "
40 ?" | DICE GAME "
50 ?" | * * | "
60 ?" | ANDREW VIRNULS"
70 ?" ~~~~~ "
90 A=0
100 B=0
110 ?:"?" " PLAYER ONE"
120 ?:"WHAT IS YOUR NAME"
130 INPUT N$
140 ?:"N$" WHAT IS YOUR LUCKY NUMBER"
150 INPUT L
160 S=RND(-L)
170 FOR N=1 TO 500: NEXT N
180 PUT 31
190 ?:"?" " PLAYER TWO"
200 ?:"WHAT IS YOUR NAME"
210 INPUT NA$
250 ?:"NA$" WHAT IS YOUR LUCKY NUMBER"
260 INPUT M
270 T=RND(-M)
275 ?:"N$,NA$"
280 FOR K=1 TO 10
290 X=INT(6*RND(1))+1
300 A=A+X
310 Y=INT(6*RND(1))+1
320 B=B+Y
330 ?X,Y
335 ?:" THROW NUMBER ";K
340 IF X>Y THEN ?N$" WON THAT ROUND"
350 IF X<Y THEN ?NA$" WON THAT ROUND"
355 FOR I=1 TO 800: NEXT I
360 NEXT K
370 ?:"?" TOTAL "
380 ?:"A,B"
390 IF A>B THEN ?:"N$" WON THAT GAME"
400 IF B>A THEN ?:"NA$" WON THAT GAME"
405 IF A=B THEN ?:"" THE GAME IS A DRAW"
410 PRINT:PRINT " ANOTHER GAME (Y/N)"
420 INPUT AG$
430 IF AG$="Y" OR AG$="YES" THEN PUT 31:GOTO 10
440 PRINT:PRINT " GOODBYE !!!"
450 END

```

goal

RUN

```

| * * |
| * * |
| * * |
| * * |
~~~~~
TEN THROW
DICE GAME
ANDREW VIRNULS

```

PLAYER ONE

WHAT IS YOUR NAME
? ANDREW

ANDREW WHAT IS YOUR LUCKY NUMBER
? 6

PLAYER TWO

WHAT IS YOUR NAME
? PAUL

PAUL WHAT IS YOUR LUCKY NUMBER
? 5

ANDREW	PAUL
4	3

THROW NUMBER 1
ANDREW WON THAT ROUND
1 2

THROW NUMBER 2
PAUL WON THAT ROUND
2 6

THROW NUMBER 3
PAUL WON THAT ROUND
4 2

THROW NUMBER 4
ANDREW WON THAT ROUND
2 2

THROW NUMBER 5
2 2

THROW NUMBER 6
6 3

THROW NUMBER 7
ANDREW WON THAT ROUND
2 1

THROW NUMBER 8
ANDREW WON THAT ROUND
3 4

THROW NUMBER 9
PAUL WON THAT ROUND
5 1

THROW NUMBER 10
ANDREW WON THAT ROUND

TOTAL

31 26

ANDREW WON THAT GAME

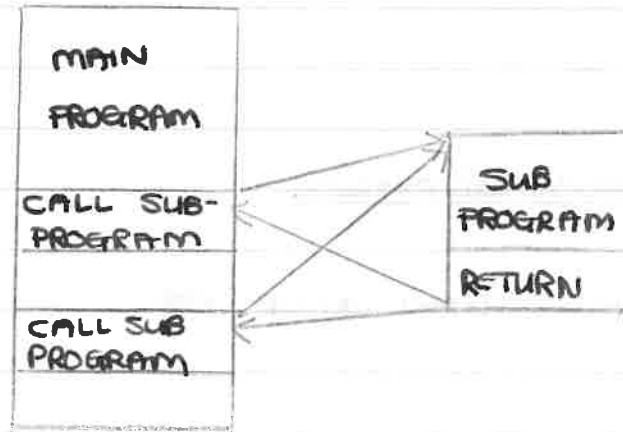
ANOTHER GAME (Y/N)
? N

GOODBYE !!!

Ready:

Subprogram

A subprogram is part of a main program which performs a specific function. The control of the subprogram is determined by a part of the main program and transferred to a subprogram is called calling up that subprogram. At the end of the subprogram routine control is transferred back to the point from which the subprogram is called.



Statements and flow chart

e.g

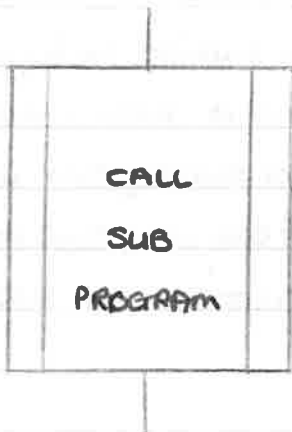
Main program

Subprogram

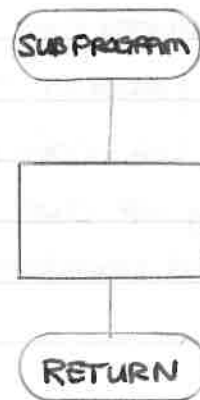
```
100 REM MAIN PROGRAM
  |
  |
  |
  |
150 GOSUB 500
155 REM
  |
  |
180 GOSUB 500
```

```
500 REM SUBPROGRAM
505 . . . . .
  |
  |
  |
  |
  |
500 RETURN
```

Main Flowchart



Subprogram Flowchart



Conditional Subprogram

50 IF THEN GOSUB 100

eg

If the paint required to cover a wall is calculated using litres per metre squared it would be a good idea to input height and length of the wall in metres. However if these values were input in feet a subroutine could be used to convert feet to metres

eg

```
50 IF N$ = "F" THEN GOSUB 100
:
:
100 REM SUBPROGRAM
110 LET D = D / 3.2808
120 RETURN
:
```

Where N\$ = choice input of metres or feet

F = feet

D = Height (or length) in feet.

Multiway Branch to a subprogram

If J has the value 1 the subroutine starting at line 200 is called.
If J=2 the subroutine at line 300 is called etc.

50 ON J GOSUB 200, 300, 400, 500

If J=5 an error will be given

eg If the computer is used as a ready reckoner ie to perform simple arithmetic operations (plus, minus, multiply, divide) then a different subroutine could be called depending on the operation required. This type of routine is referred to as choosing a menu.

1 - Addition

J=1; Subroutine 200 = Addition program

Mathematical Functions

1. RND (x) , INT (x) - see previous work
2. ABS (x) - returns absolute value of number
eg ABS (-9) = 9 ABS (4) = 4.
3. SIGN (x) - tests for sign of number
eg SIGN (x) = +1 if x is positive
" " = 0 " " " zero
" " = -1 " " " negative
4. SQR (x) provides the square root of a positive number
eg SQR (16) = 4
5. SIN (x), COS (x), TAN (x)

Angle must be in radians

π rads = 180°

Homework Problems

Write a fully documented to enable you carry out physics or chemistry calculation by supply numbers with are the substituted into the appropriate formula. In the main program display a menu of the available formulae one of which is called by a suitable input. In the subprograms for each formula display the formula together with definitions and units for all of the variables, and prompt for the input of the required variables. Evaluate the formula and display the result before returning to the menu.

N.B

1. In all programs define variables
2. In flow chart and algorithm avoid use of BASIC statements (must be language independent)
3. Neat presentation important, use all headings in correct order
4. Listing from computer essential only $\frac{1}{2}$ marks available if non-given
5. Title must be short and to the point.

Character Handling functions

LEFT\$(X\$,N)

This returns the first N characters from a character string X\$

eg

If X\$ is letters of the alphabet eg ABCDE etc

LEFT\$(X\$,3) returns ABC

RIGHT\$(X\$,N)

This returns the last N characters of character string X\$

eg

RIGHT\$(X\$,N) returns XYZ

MID\$(X\$,M,N)

This returns the N characters from the string X\$ starting at position M

e.g

MID\$(X\$,3,2) returns CD

N.B

Spaces are counted as characters in a string

LEN(X\$)

This returns the length of string X\$

eg

LEN(X\$) returns 26 (if there are no spaces in the string)

User defined functions.

e.g 1) 110 DEF FNS(X) = X * X

This defines a function S which returns the square of the argument X.

⋮

150 LET A = 3.2

160 LET B = FNS(A)

Assigns to variable B value 10.24

2) 120 DEF FNR(X) = INT(X + 0.5)

Defines a function R which rounds up X to the next highest integer

⋮

150 LET A = 3.35

160 LET B = FNR(A)

B=4

3) 110 DEF FNJS(X\$) = MID\$(X\$, 1, 1)

This defines function JS which produces a defined middle section of X\$

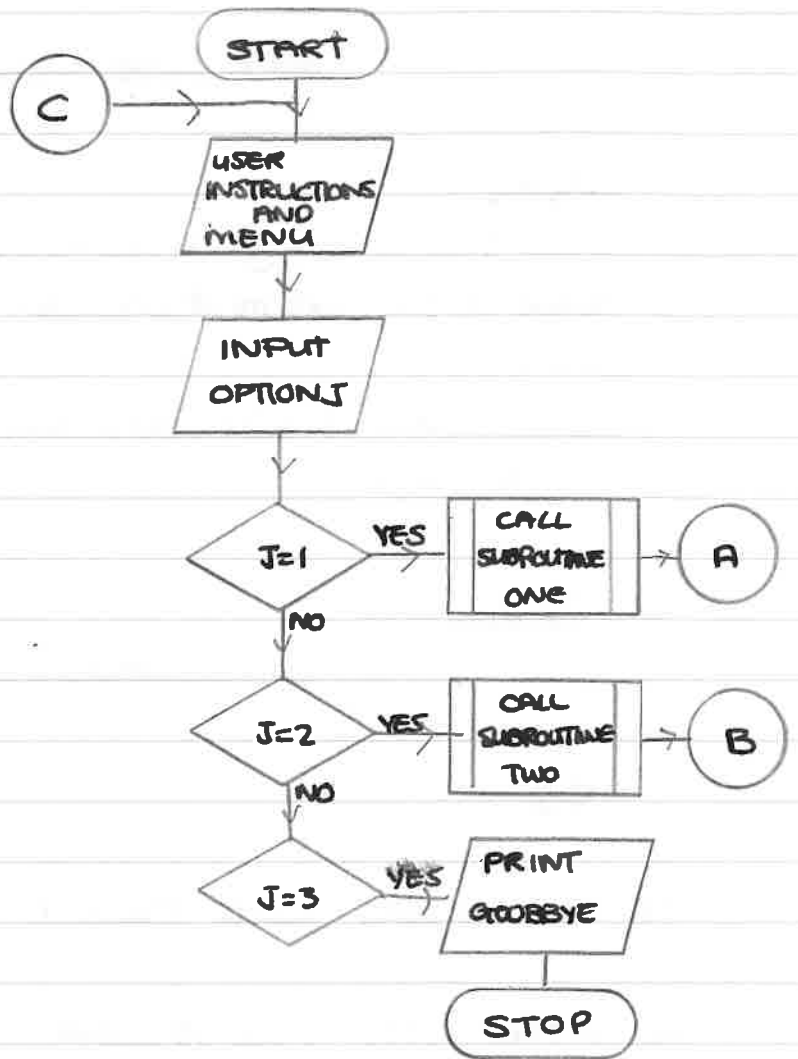
Subroutine Flowchart

Program

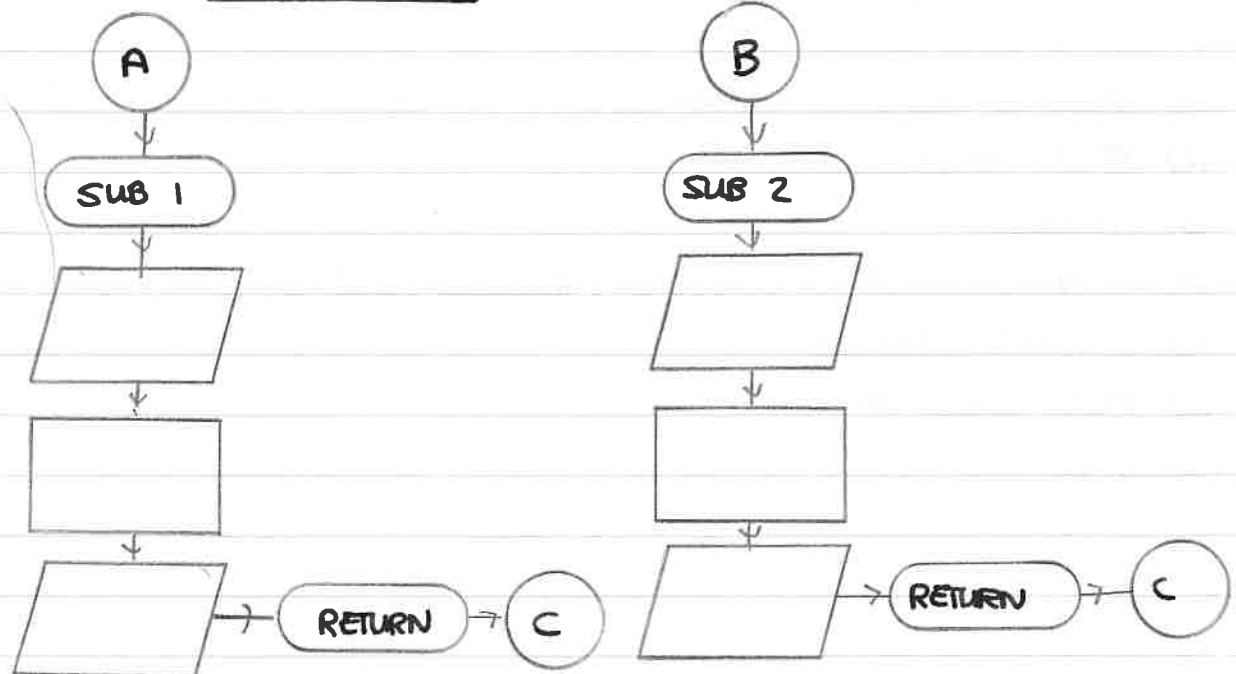
```

10 PRINT "HEADINGS"
20 ? "1 - ADD"
30 ? "2 - SUBTRACT"
40 ? "3 - EXIT"
50 ? "INPUT OPTIONS"
60 INPUT J
70 ON J GOSUB 100, 200, 300
80 GOTO 10
100 REM SUBROUTINE 1 - ADD
    .
    .
150 RETURN
200 REM SUBROUTINE 2 - SUBTRACT
    .
    .
250 RETURN
300 ? "GOODBYE"
310 END
    
```

main Flowchart



Subroutines



Homeworks

Document a program on string handling using the functions LEFT\$, RIGHT\$, MID\$, LEN, CHR\$, or ASC

GET statement

Defn

This is a single character input routine which returns the ASCII value of the next input character

e.g. GET () will return the next character input from the keyboard

Timing

The input can be optionally timed for interactive computing.

e.g

40 A = GET (500) waits 5 seconds

40 A = GET (1000) waits 10 seconds

Sample Program

```
110 PRINT CHR$(31) (PUT 31)
120 PRINT "ANOTHER GO (Y/N)?"
130 A = GET ( )
140 IF A = 89 THEN . . . . . (Tests for Yes)
150 IF A < 78 THEN 130
160 PRINT "THE END"
170 END.
```


RUN

Ready:
RUN

ENTER STRING

YOU ENTERED GOODBYE

R
PRESS ANY KEY
U

Ready:

GET\$()

This used when you want to return the actual character entered

```
40 A$ = GET$( )
```

```
50 PRINT A$
```

$A\$ = GET$() = A\$ = CHR$(GET())$

Homework

1) 10	PUT 31	10	PUT 31
20	FOR I = 1 TO 5	20	A\$ = ""
30	? "PRESS ANY KEY"	30	? "ENTER STRING"
40	A\$ = GET\$()	40	A = GET()
50	? A\$	50	IF A = 13 THEN 80
60	NEXT I	60	A\$ = A\$ + CHR\$(A)
70	END	70	GOTO 40
		80	?
		90	? "YOU ENTERED "; A\$

TRY ONE OF THESE PROGRAMS find out what it does and modify it to make your own program (a game with user instructions, timing etc)

```

10 REM music
20 PUT 31
30 ??:?:?:?:?:?:?:? " Press:- "
40 ?:" 1 = Middle C "
50 ? " 2 = D "
60 ? " 3 = E "
70 ? " 4 = F "
80 ? " 5 = G "
90 ? " 6 = A "
100 ? " 7 = B "
110 ? " SPACE BAR - SPECIAL EFFECT "
115 ??:?:?:? "NOTE:"

```

```

120 ENVELOPE 1, 9, 8, 7, 6, 5, 4, 3, 2, 1, 3, 5, 7, 9
130 *FX 12, 1
140 *FX 11, 1
150 REPEAT
160 A = GET()
170 ?TAB (5, 20); ""
180 IF A = 49 THEN SOUND 1, -15, 4, 4 : ?TAB (5, 20); "C"
190 IF A = 50 THEN SOUND 1, -15, 8, 4 : ?TAB (5, 20); "D"
200 IF A = 51 THEN SOUND 1, -15, 12, 4 : ?TAB (5, 20); "E"
210 IF A = 52 THEN SOUND 1, -15, 16, 4 : ?TAB (5, 20); "F"
220 IF A = 53 THEN SOUND 1, -15, 20, 4 : ?TAB (5, 20); "G"
230 IF A = 54 THEN SOUND 1, -15, 24, 4 : ?TAB (5, 20); "A"
240 IF A = 55 THEN SOUND 1, -15, 28, 4 : ?TAB (5, 20); "B"
250 IF A = 32 THEN SOUND 1, 1, 50, 20 : ?TAB (5, 20); "S"
260 UNTIL FALSE

```

The program turns the keyboard into a musical keyboard with "1" being pressed for middle C, "2" for D, "3" for E up the scale to B. Pressing the space bar produces a special effect.

Results

Press: -

1 - middle C

2 - D

3 - E

4 - F

5 - G

6 - A

7 - B

SPACE BAR - SPECIAL EFFECT

NOTE: A

8
10.

Woodhouse Buggy

Control Relays



Used to represent our binary number in an 8 bit word

Signals out - user port

This uses address $\&11D$ (ampersand)
(hexadecimal number)

BASIC

- 10 P = $\&11D$
- 20 OUT P, A - decimal value of "switched relays"

Sample Program

- 10 P = $\&11D$
- 20 D\$ = GET\$(1)
- 30 IF D\$ = "F" THEN A = 97
- 40 IF D\$ = "B" THEN A = 82
- 50 IF D\$ = "L" THEN A = 228
- 60 IF D\$ = "R" THEN A = 216
- 70 IF D\$ = "S" THEN A = 0
- 80 OUT P, A
- 90 GOTO 20

File Handling

Statements

CREATE #10, F\$ - create a file called F\$, and open it for writing.

PRINT #10, A, B\$ - write a record to a file, containing data items A and B\$.

OPEN #10, F\$ - open file F\$ for reading.

INPUT #10, A, B\$ - read a record from a file, containing data items A and B\$.

CLOSE #10 - close a file at the end of writing and reading.

Exercise 120 Qu3.

Write a program using the pupil records file, to accept as input the name of a particular pupil, and locate and print the record for that pupil. If the pupil name cannot be found, print a suitable message.

```
10 REM FILE READER.
20 INPUT "NAME OF FILE TO BE READ", F$.
30 OPEN #10, F$.
40 INPUT "NAME OF PUPIL", C$.
50 INPUT #10, N$, S$, B$, M$.
60 IF N$ = C$ THEN 110
70 IF N$ = "*" THEN 90
80 GOTO 50
90 PRINT "FILE NOT FOUND"
100 GOTO 10
110 PRINT "NAME"; N$; "SEX"; S$; "DATE OF BIRTH"; B$; "FORM"; M$;
120 GOTO 10
```



① Introduction To Graphics.

LOW RESOLUTION GRAPH TEXT.

Enter graphics mode by typing
GRAPH [return] or in a program by
using $\text{\textcircled{G}}$ GRAPH.

The screen will divide into 2 areas :-

- 20 lines at top for graphics
- 4 lines at the bottom for text.

The graphics area may be thought of as
being 20 lines of 40 characters. Each rectangle
is a rectangle of 6 squares. 

Co-ordinates



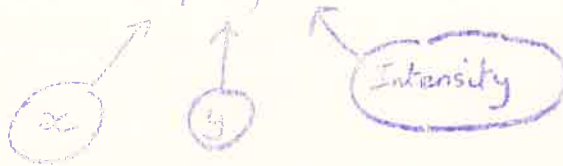
To revert to normal display type TEXT (return)
or $\text{\textcircled{T}}$ TEXT.

2

PLOT

The plot command is used to plot points, characters or strings anywhere in the graphics area.

e.g. 2ϕ PLOT 40, 60, 2



Find out what happens when the 2 is replaced with 1 and then 0.

Plotting strings.

```

1ϕ A$ = "HELLO"
2ϕ B$ = "FRED"
3ϕ GRAPH
4ϕ PLOT 10, 50, A$
5ϕ PLOT 15, 40, B$

```

Example - This program plots STRINGS read from DATA statements.

```

1ϕ GRAPH
2ϕ X = 10
3ϕ Y = 55
4ϕ FOR I = 1 TO 10
5ϕ READ A$(I)
6ϕ PLOT X, Y, A$(I)
7ϕ Y = Y - 5
8ϕ NEXT I
9ϕ DATA ONE, TWO, THREE, FOUR etc.

```


③ DRAWING LINES.

- The LINE command draws a straight line from the co-ords of the last PLOT command (or at the end of the last line), to the position stated.

e.g. 1Ø GRAPH
2Ø PLOT 10,10,2
3Ø LINE 70,10,2
4Ø LINE 70,50,2
5Ø LINE 10,50,2
6Ø LINE 10,10,2



Modify the program to change the rectangle from white to grey.

Draw diagonals linking the corners.

PLOTTING characters by Number.

e.g. PLOT 20,20,56

ASCII
CODE

Try this program

```
1Ø GRAPH
2Ø FOR I = 0 TO 255
3Ø PLOT 20,20,I
4Ø NEXT I
5Ø TEXT.
```

To slow the program down introduce a DELAY

```
35 FOR DELAY = 1 TO 2ØØ
37 NEXT DELAY
```


7

GRAPHICS AND MOTION

(ANIMATION)

Example

- 1φ $x = \phi : y = 3\phi$
- 2φ GRAPH
- 3φ PLOT $x, y, 2$
- 4φ PLOT x, y, ϕ
- 5φ $x = x + 1$
- 6φ IF $x \leq 79$ THEN 3φ
- 7φ TEXT

Moves a dot across screen.

To slow motion down, introduce a delay

- 85 FOR DELAY TO 200
- 87 NEXT DELAY

POINT

The point command allows each pixel on the screen to be tested to determine what has been plotted

Example

- 1φ GRAPH
- 2φ PLOT 0,0,2
- 3φ LINE 79,0
- 4φ LINE 79,59
- 5φ LINE 0,59
- 6φ LINE 0,0
- 7φ $x = 40 : y = 30 : h = 1 : v = 1$
- 8φ PLOT x, y, ϕ
- 9φ $x = x + h$
- 10φ $y = y + v$
- 11φ PLOT $x, y, 2$
- 12φ IF POINT $(x, y + v) = 2$ THEN $v = -v$
- 13φ IF POINT $(x + h, y) = 2$ THEN $h = -h$
- 140 GOTO 80

Creation of a border and then moves on dot across the screen when border reached direction is reversed.

Type in,
Try,
MODIFY

Algorithms

Defn. An algorithm is a step by step procedure for solving a particular problem. Algorithms can range from the very simple like making a cup of tea to the extremely complex like launching the Space Shuttle.

Example

Write a programme to read in a bank account number, the amount in the account (the balance) and then ~~ten~~ deposits or with draws each with a date

Withdrawals are entered as negative numbers and so both are added to the balance to get the new balance.

A bank statement is printed showing the account number and original balance and the columns for the date, deposit or withdrawal and the new balance.

Description

This section must describe the programme so that any user would be able to understand the logic of the problem.

Design of Input (Data or Input information)

The section defines the use of data in the programme

Formulae

This section represents any form of calculation performed in the programme

ie LET statements

Algorithm

1. Input account number
2. Input information

3. Calculate new balance
4. Output result.

1st refinement

- 1.1 Input name
- 1.2 Input account number 1.3 Input Date
- 2.1 Input Old Balance
- 2.2 Input Withdrawals or deposits (date of transaction)
- 3.1 Calculate new balance
- 4.1 Output results

2nd refinement

- 1.1 Input name
- 1.2 Input account number
- 1.3 Input Date
- 2.1 Input Old balance
- 2.2 Input withdrawals or deposits (date of transaction)
- 3.1 Find old balance
- 3.2 Add withdrawal and deposit figures to old balance to give new balance
- 4.1 Current date printed.. Name and account N^o
- 4.2 Table Headings printed.
- 4.3 Table containing old balance, withdrawals, deposits, date, new balance

Flow chart

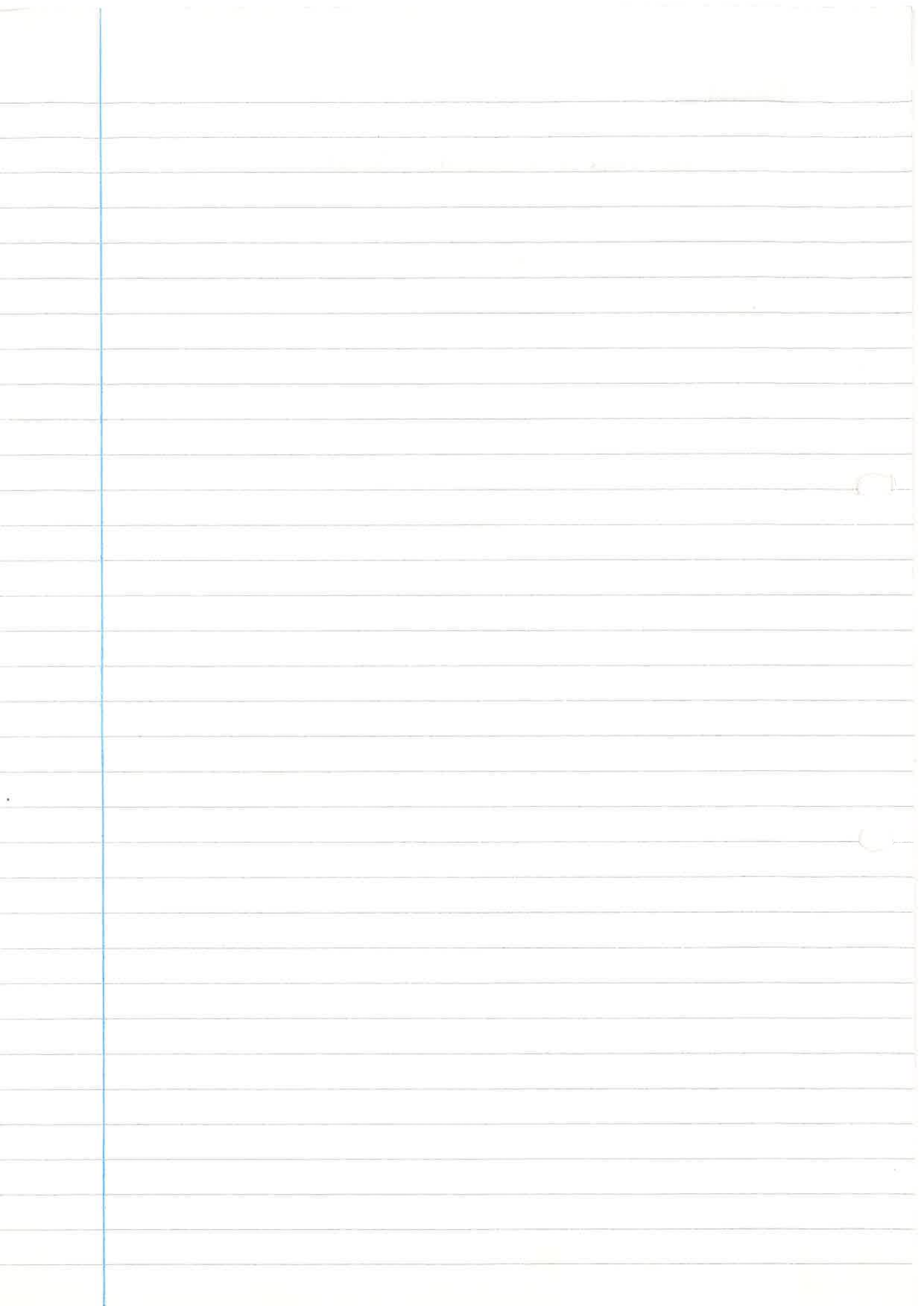
Using specified symbols translate algorithm into flowchart.

Programme

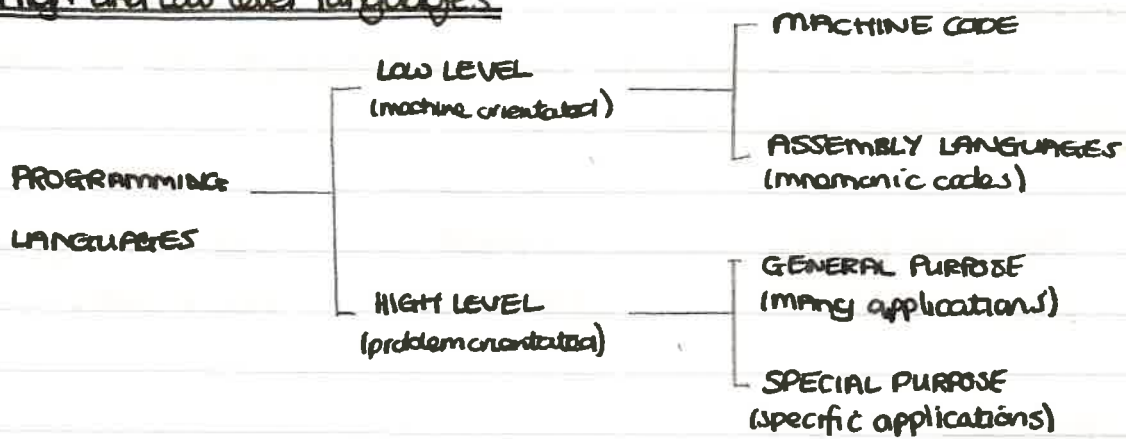
Listing from computer-

Results

This must be a computer printout.



High and Low level languages



Definitions

1. High level languages

Human orientated languages that can be used on many different machines. It has to be decoded and is slow and takes a lot of memory e.g. BASIC, Lisp, Fortran.

2. Low level Languages

Machine orientated languages which are different for every machine. Hard to use, but faster and take less memory. The machine does not have to decode it.

3. Source code / Program

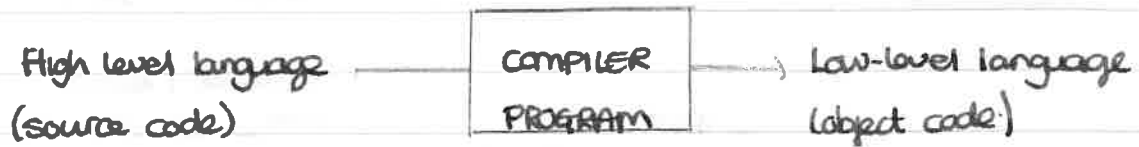
A program that is in a high level language which is translated into a low-level language ie by a compiler.

4. Object code / Program

The machine code that will run on any machine. It is the translated form of a high level language.

5 Compiler

A program which converts high level language into low level language (Source code to object code)



6 Interpreter

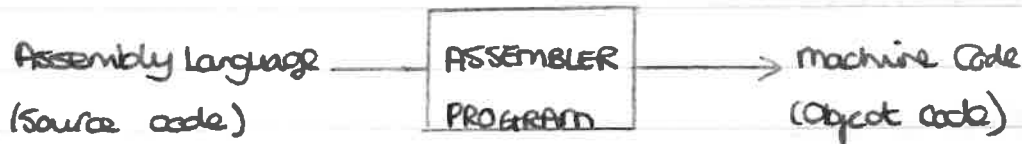
This converts source code to object code one line at a time

7 Translator

This changes any language to another. It is a general term.

8 Assembler

A program which converts assembly language into machine code.



High level Language

```
50 READ A,B
60 LET C=A+B
70 PRINT C
:
150 DATA 1,2
```

Assembly language

Format

Label	Function / Operation code	operation / address
-------	------------------------------	------------------------

Mnemonic Function / Operation code

StH p 40 fig 2.4

StH p 40 fig 2.5 - reused version

Label	Function Code	Address
5	IN	
6	STO	33
7	IN	
8	ADD	33
9	OUT	

Machine Code

Label	Function Code	Address
0101	0111	-
0110	0110	00100001
0111	0111	-
1000	0001	00100001
1001	1000	-

Symbolic Addressing

Symbolic addresses are used so that you don't have to remember the address where numbers are stored

eg STO NUM1
STU NUM2
STO RES1 etc

StH p 43 fig 2.9.

Homework

Write the program on p40 Sth in Both machine code and then in Assembly code but extending the program to Subtract the second number from the first as well as adding and multiplying

use binary fig 2.8 dont use function code use symbolic names.

translate into decimal, then binary machine code.

<u>Mnemonics</u>			→	<u>Decimal</u>			→	<u>Binary (machine code)</u>		
1	IN		=	1	07		=	000001	00111	
2	STO	NUM1	=	2	06	32	=	000010	00110	00100000
3	IN		=	3	07		=	000011	00111	
4	STO	NUM2	=	4	06	33	=	000100	00110	00100001
5	ADD	NUM1	=	5	01	32	=	000101	00001	00100000
6	STO	RES1	=	6	06	34	=	000110	00110	01000010
7	LDA	NUM1	=	7	05	32	=	000111	00101	01000000
8	MULT	NUM2	=	8	03	33	=	001000	00011	01000001
9	STO	RES2	=	9	06	35	=	001001	00110	01000011
10	LDA	NUM1	=	10	05	32	=	001010	00101	01000000
11	SUB	NUM2	=	11	02	33	=	001011	00010	01000001
12	STO	RES3	=	12	06	36	=	001100	00110	01000100
13	LDA	RES1	=	13	05	34	=	001101	00101	01000010
14	OUT		=	14	08		=	001110	01000	
15	LDA	RES2	=	15	05	35	=	001111	00101	01000011
16	OUT		=	16	08		=	010000	01000	
17	LDA	RES3	=	17	05	36	=	010001	00101	01000100
18	OUT		=	18	08		=	010010	01000	

good work
A-