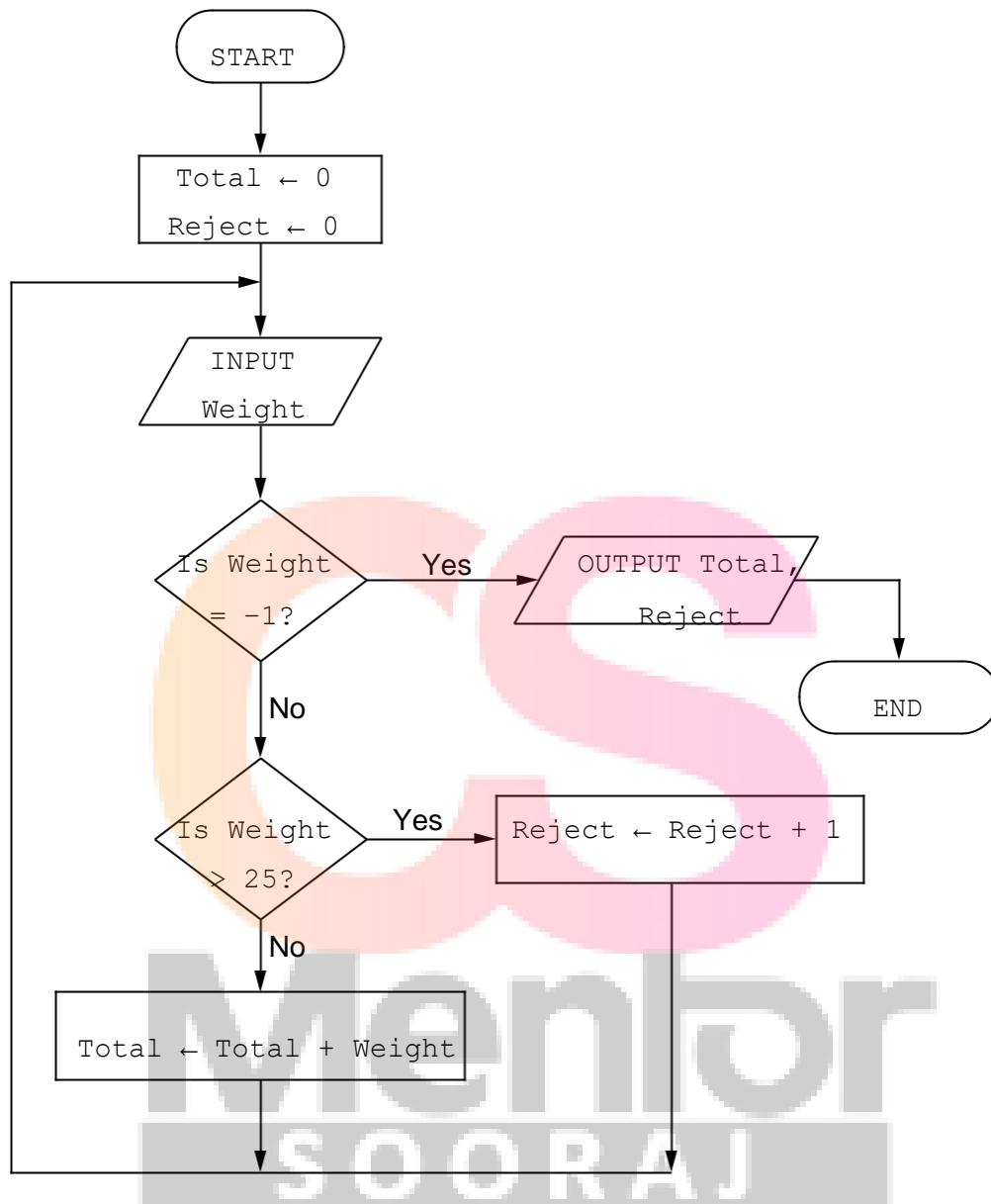# HANDOUT FOR CHAPTER 9

PROBLEM SOLVING AND DESIGN

Past questions

© UCLES 2015 -2020

1   The flowchart below inputs the weight of a number of parcels in kilograms. Parcels weighing more than 25 kilograms are rejected. A value of –1 stops the input.

The following information is output: the total weight of the parcels accepted and number of parcels rejected.
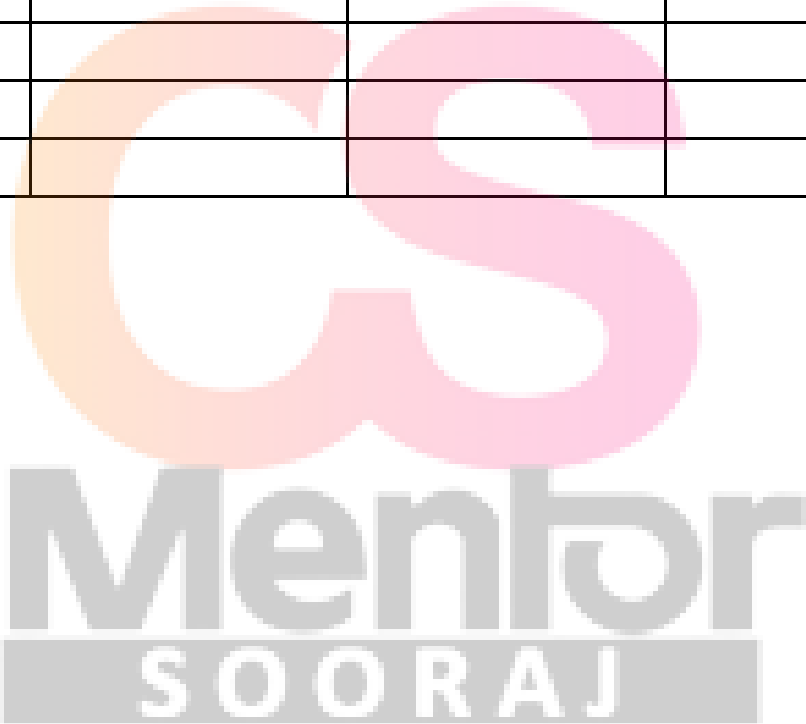
```
                        ┌─────────────┐
                        │    START     │
                        └──────┬──────┘
                               │
                        ┌──────▼──────┐
                        │  Total ← 0   │
                        │  Reject ← 0  │
                        └──────┬──────┘
                               │
                        ┌──────▼──────┐
                        │   INPUT      │
                        │   Weight     │
                        └──────┬──────┘
                               │
                          ╱────▼────╲        Yes    ┌──────────────────┐
                         ╱ Is Weight  ╲─────────────│  OUTPUT Total,   │────┐
                         ╲  = -1?     ╱              │     Reject       │    │
                          ╲────┬────╱               └──────────────────┘    │
                               │ No                                    ┌─────▼─────┐
                               │                                       │    END     │
                          ╱────▼────╲        Yes    ┌──────────────────┐
                         ╱ Is Weight  ╲─────────────│ Reject ← Reject + 1│
                         ╲  > 25?     ╱              └──────────────────┘
                          ╲────┬────╱
                               │ No
                        ┌──────▼──────────────────┐
                        │ Total ← Total + Weight   │
                        └──────┬──────────────────┘
```

Complete the trace table for the input data:

1.8, 26.0, 7.0, 11.3, 10.0, 2.5, 25.2, 5.0, 19.8, 29.3, –1

Total	Reject	Weight	OUTPUT

[5]

2 Read this section of program code that should input 10 positive numbers and then output the smallest number input.

```
1    Small = 0

2    Counter = 0

3    REPEAT

4        INPUT Num

5        IF Num < Small THEN Num = Small

6            Counter = Counter + 1

7        PRINT Small

8    UNTIL Counter < 10
```
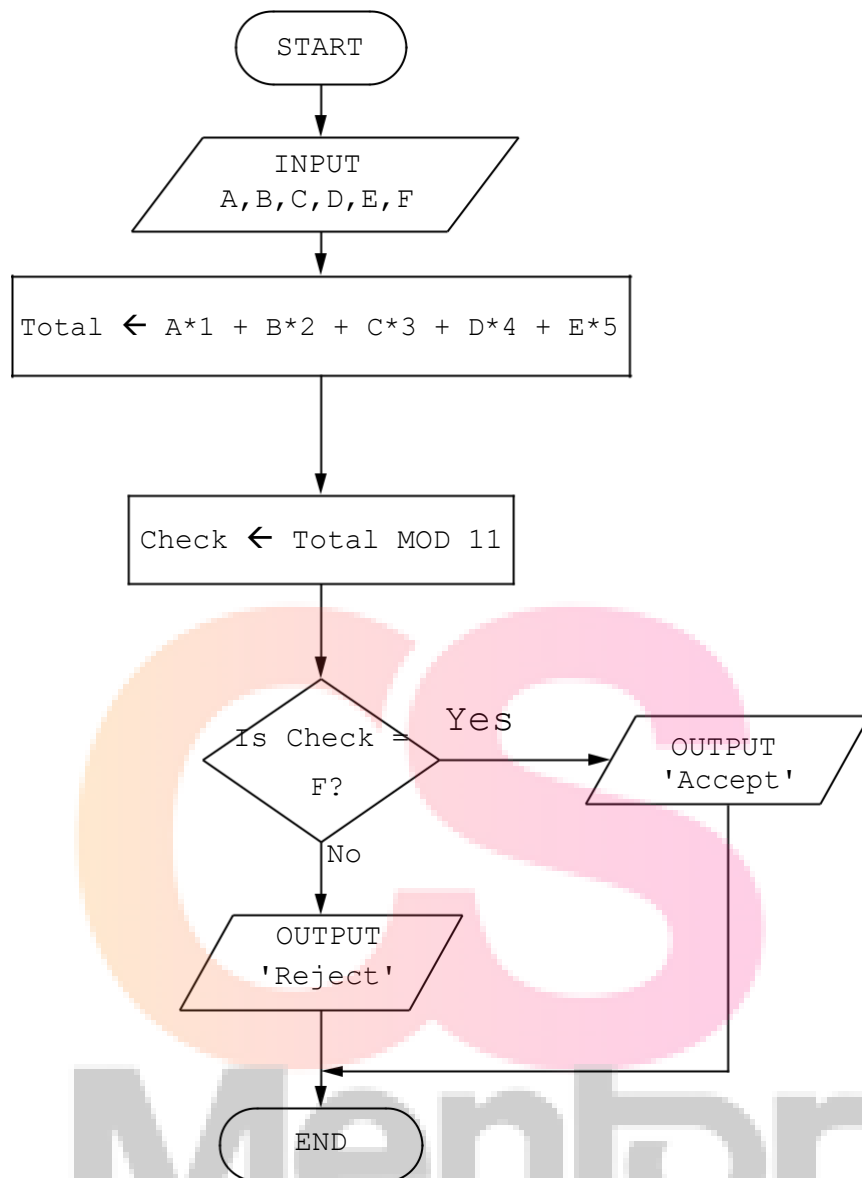
There are **four** errors in this code.

Locate these errors and suggest a corrected piece of code for each error.

1 ....

....

2 ....

....

3 ....

....

4 ....

....[4]

Page | 4

3 **(a)** The flowchart below inputs six single digit numbers. The predefined function `MOD` gives the value of the remainder, for example, $Y \leftarrow 10\ MOD\ 3$ gives the value $Y = 1$

```
                        ( START )
                            |
                            v
                   /  INPUT          /
                  /   A,B,C,D,E,F   /
                            |
                            v
        +---------------------------------------+
        | Total <- A*1 + B*2 + C*3 + D*4 + E*5  |
        +---------------------------------------+
                            |
                            v
              +--------------------------+
              | Check <- Total MOD 11    |
              +--------------------------+
                            |
                            v
                          /    \                    Yes    /  OUTPUT   /
                    Is Check =  --------------------------/  'Accept' /
                          F?                                    |
                            \    /                              |
                            |No                                 |
                            v                                   |
                    /  OUTPUT   /                               |
                   /  'Reject' /                                |
                            |                                   |
                            v<----------------------------------+
                        ( END )
```

Complete a trace table for each of the two sets of input data.

Set 1   5,2,4,3,1,5

Set 2   3,2,1,0,7,3

Trace table set 1   5,2,4,3,1,5

A	B	C	D	E	F	Total	Check	Output

Trace table set 2   3,2,1,0,7,3

A	B	C	D	E	F	Total	Check	Output

[4]

(b) State the purpose of the flowchart in **part (a)**.

....

....[1]

(c) Identify a problem with this flowchart and explain how to correct it.

Problem ....

....

Solution ....

....

....

....[3]

4 Read this section of program code that should input 30 positive numbers and then output the largest number input.

```
1    Large = 9999

2    Counter = 0

3    WHILE Counter > 30

4    DO

5        INPUT Num

6        IF Num < Large THEN Large = Num

7            Counter = Counter - 1

8    ENDWHILE

9    PRINT Large
```

There are **four** errors in this code.

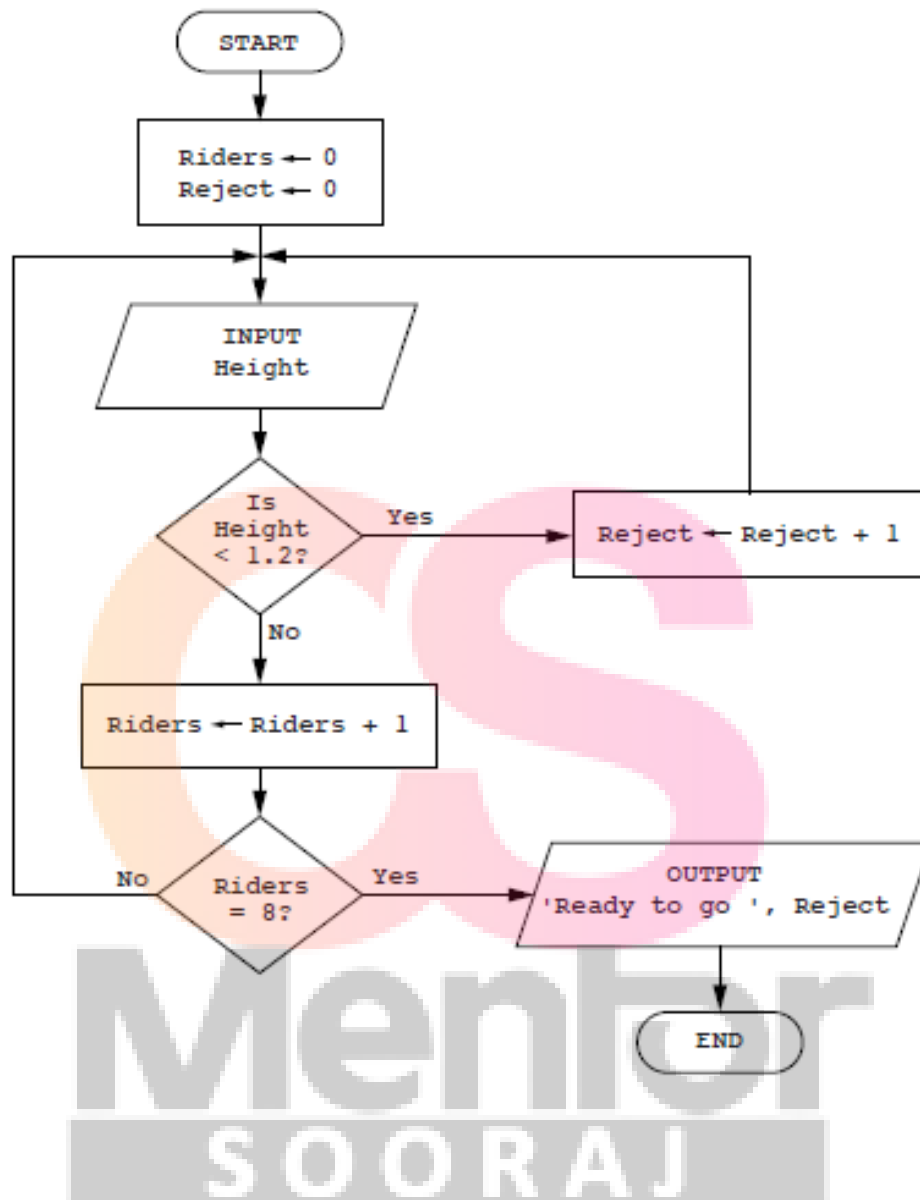Locate these errors and suggest a corrected piece of code for each error.

1 ....

....

2 ....

....

3 ....

....

4 ....

....[4]

5   The flowchart below inputs the height of children who want to ride on a rollercoaster. Children under 1.2 metres are rejected. The ride starts when eight children have been accepted.
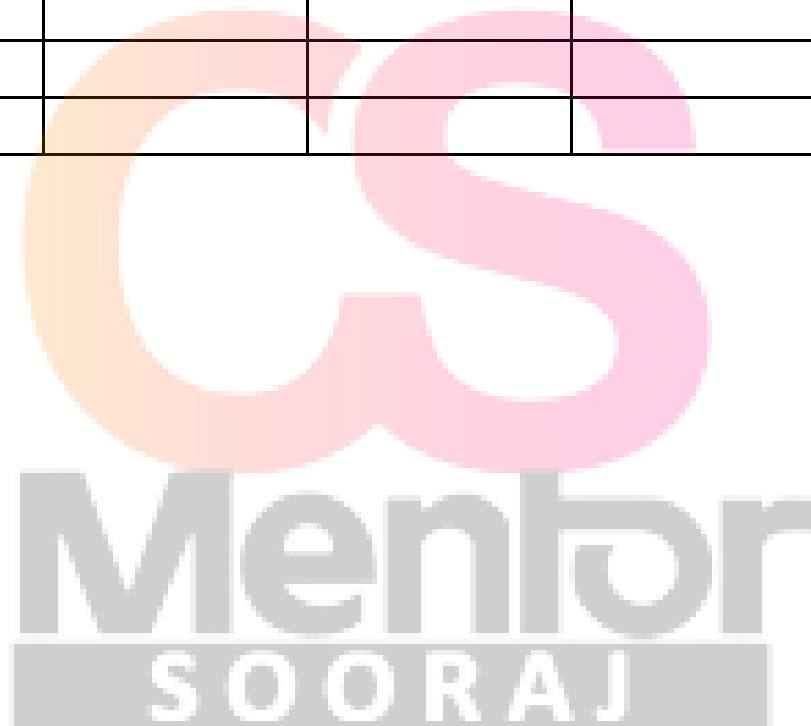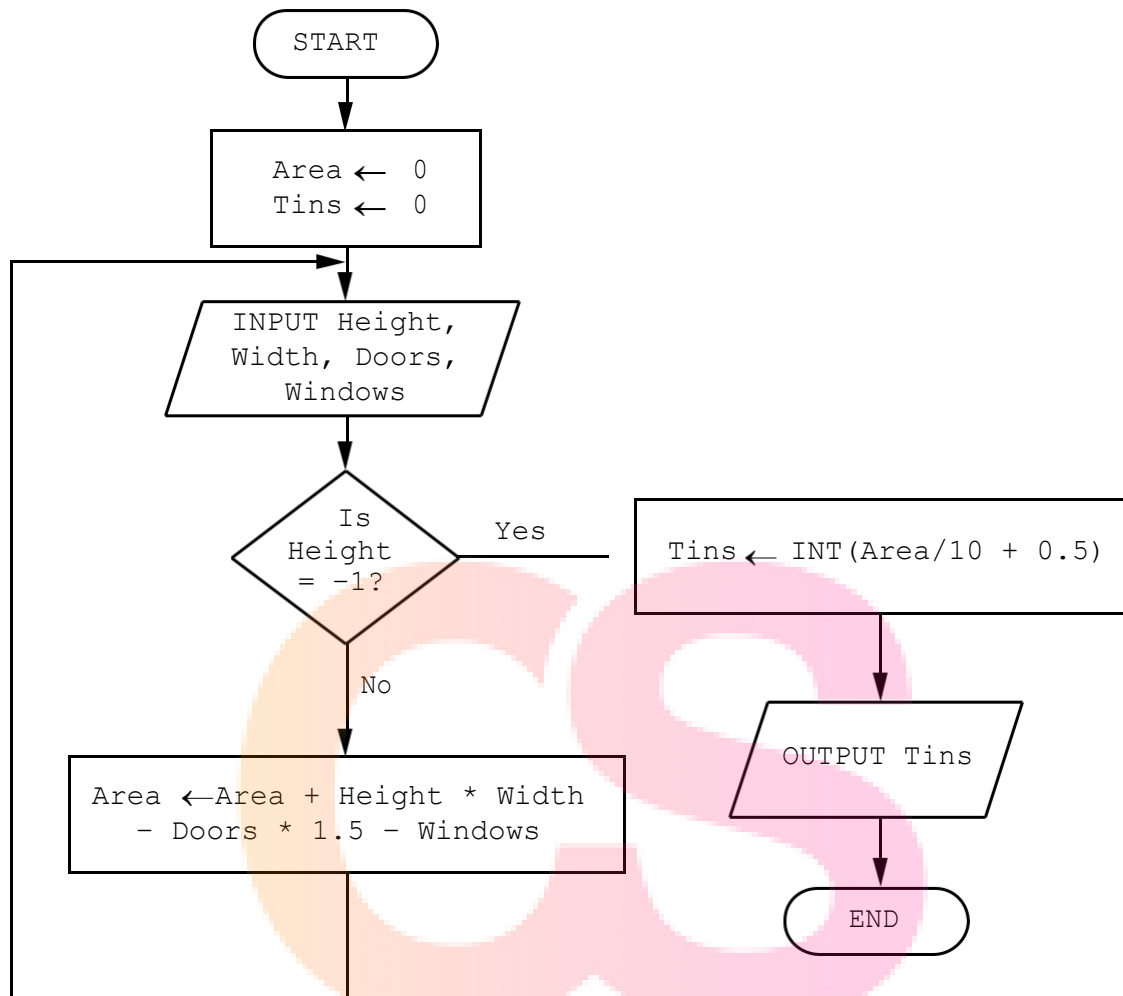
Complete the trace table for the input data:

1.4, 1.3, 1.1, 1.3, 1.0, 1.5, 1.2, 1.3, 1.4, 1.3, 0.9, 1.5, 1.6, 1.0

Riders	Reject	Height	OUTPUT

[4]

6 The flowchart below calculates the number of tins of paint required to paint walls. The flowchart inputs the height and width of a wall in metres, the number of doors and the number of windows. A value of –1 for the height stops the input.

Complete the trace table for the input data:

3, 5, 1, 0, 3, 7, 0, 0, 3, 5, 0, 3, 3, 7, 1, 1, −1, 0, 0, 0

Area	Tins	Height	Width	Doors	Windows
0	0				
13.5		3	5	1	0
34.5		3	7	0	0
46.5		3	5	0	3
65		3	7	1	1
	7	−1	0	0	0

[4]

7 Read this section of program code that inputs 10 positive numbers and then outputs the total.

```
1    Total = 0
2    Counter = 0
3    REPEAT
4        INPUT Num
5        Total = Total + Num
6        PRINT Total
7        Counter = Counter + 1
8    UNTIL Counter = 10
```

This code works, but it is inefficient.

Suggest **three** improvements that could be made.

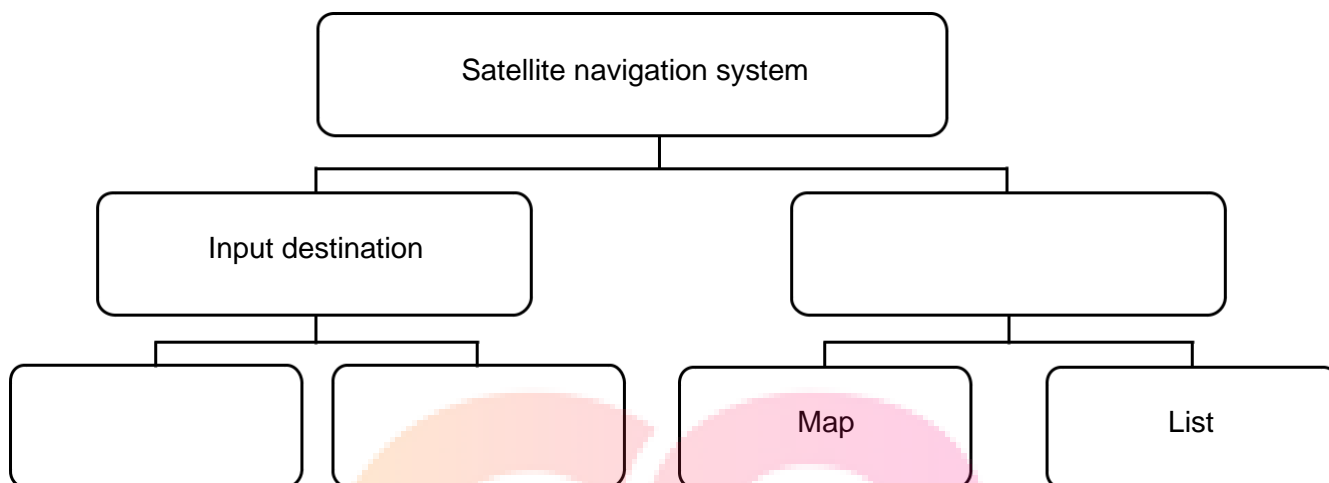1..

....

2..

....

3..

....[3]

(ii) Rewrite the program code with your improvements.

....

....

....

....

....

....

....

....

....

....[3]

8 A satellite navigation system works using destination details entered by the user, either a new destination or chosen from previously saved destinations. The satellite navigation system will then output directions to the destination in the form of either a visual map or a list of directions.

A satellite navigation system is an example of a computer system that is made up of sub-systems. This structure diagram shows some of its sub-systems.

Complete the diagram by filling in the empty boxes.

```
                    ┌────────────────────────────────┐
                    │  Satellite navigation system   │
                    └────────────────────────────────┘
              ┌────────────────────┴────────────────────┐
      ┌───────────────────┐                    ┌───────────────────┐
      │ Input destination │                    │                   │
      └───────────────────┘                    └───────────────────┘
       ┌────────┴────────┐                     ┌────────┴────────┐
  ┌─────────┐     ┌─────────┐            ┌─────────┐     ┌─────────┐
  │         │     │         │            │   Map   │     │  List   │
  └─────────┘     └─────────┘            └─────────┘     └─────────┘
```

[2]

9 For each of the **four** statements in the table, place a tick in the correct column to show whether it is an example of **validation** or **verification**.

| Statements	Validation	Verification
To automatically check the accuracy of a bar code		
To check if the data input is sensible		
To check if the data input matches the data that has been supplied		
To automatically check that all required data fields have been completed		

[4]

10 This flowchart inputs a range of temperatures in degrees Fahrenheit.

As each temperature is input, it is compared with the previous highest temperature. If it is higher than the current highest, it replaces the previous highest temperature and then it is converted to degrees Celsius.

For ease of calculation, the final step of the Fahrenheit to Celsius conversion has been approximated as division by 2.

When –1 is entered, the input process stops and the highest temperature (in both Fahrenheit and Celsius) is output.
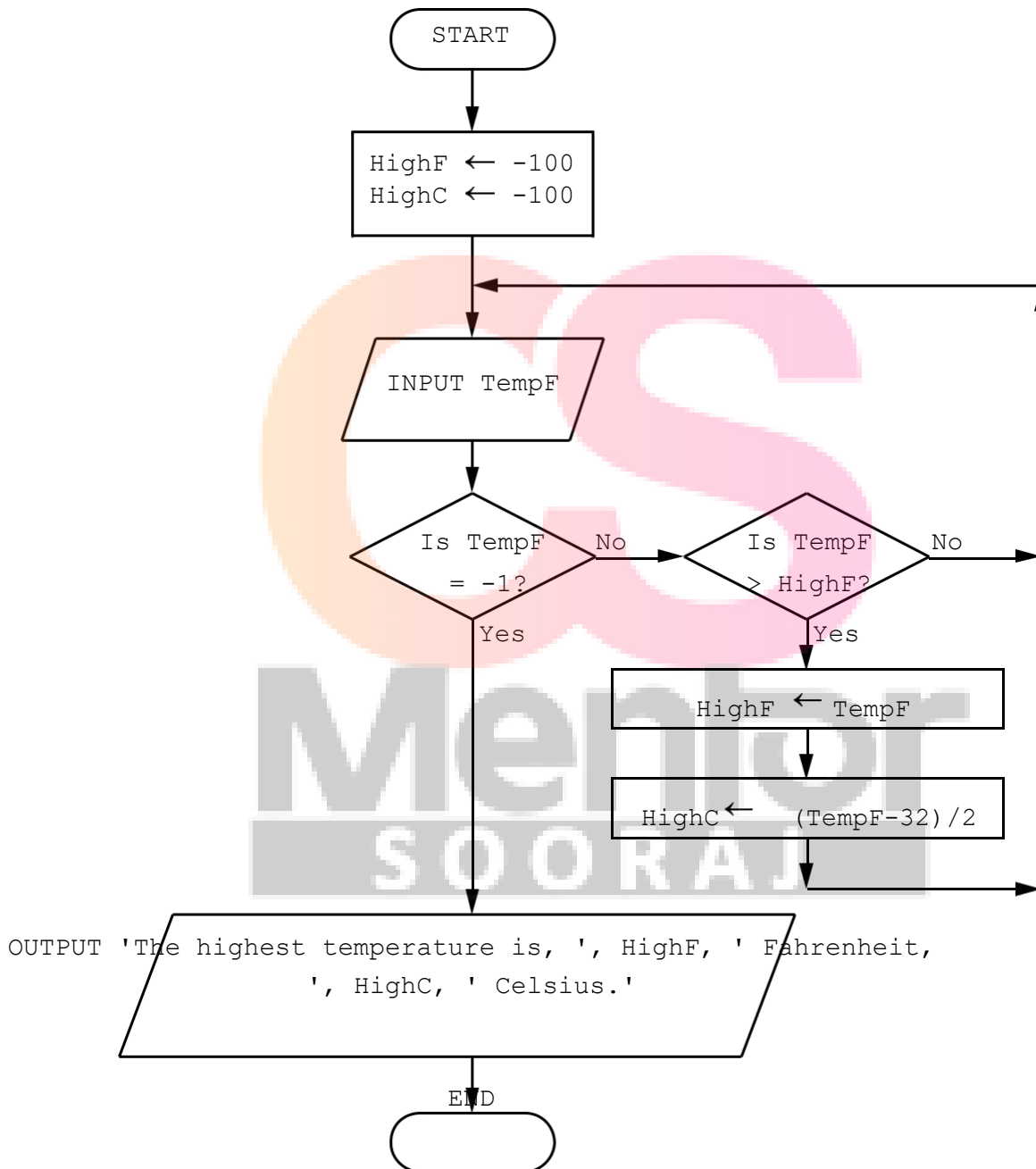
Complete the trace table for the input data:

68, 46, 50, 86, 65, 50, 40, 30, –1

HighF	HighC	TempF	OUTPUT

[5]

11 This section of program code asks for 50 numbers to be entered. The total and average of the numbers are calculated.

```
1    Total = 0
2    Counter = 50
3    PRINT 'When prompted, enter 50 numbers, one at a time'
4    REPEAT
5        PRINT 'Enter a number'
6        INPUT Number
7        Total + Number = Total
8        Number = Number + 1
9    UNTIL Counter = 50
10   Average = Number * Counter
11   PRINT 'The average of the numbers you entered is ', Average
```

There are **four** errors in this code.

State the line number for each error and write the correct code for that line.

Error 1 Line number ....

Correct code ....

Error 2 Line number ....

Correct code ....

Error 3 Line number ....

Correct code ....

Error 4 Line number ....

Correct code ....
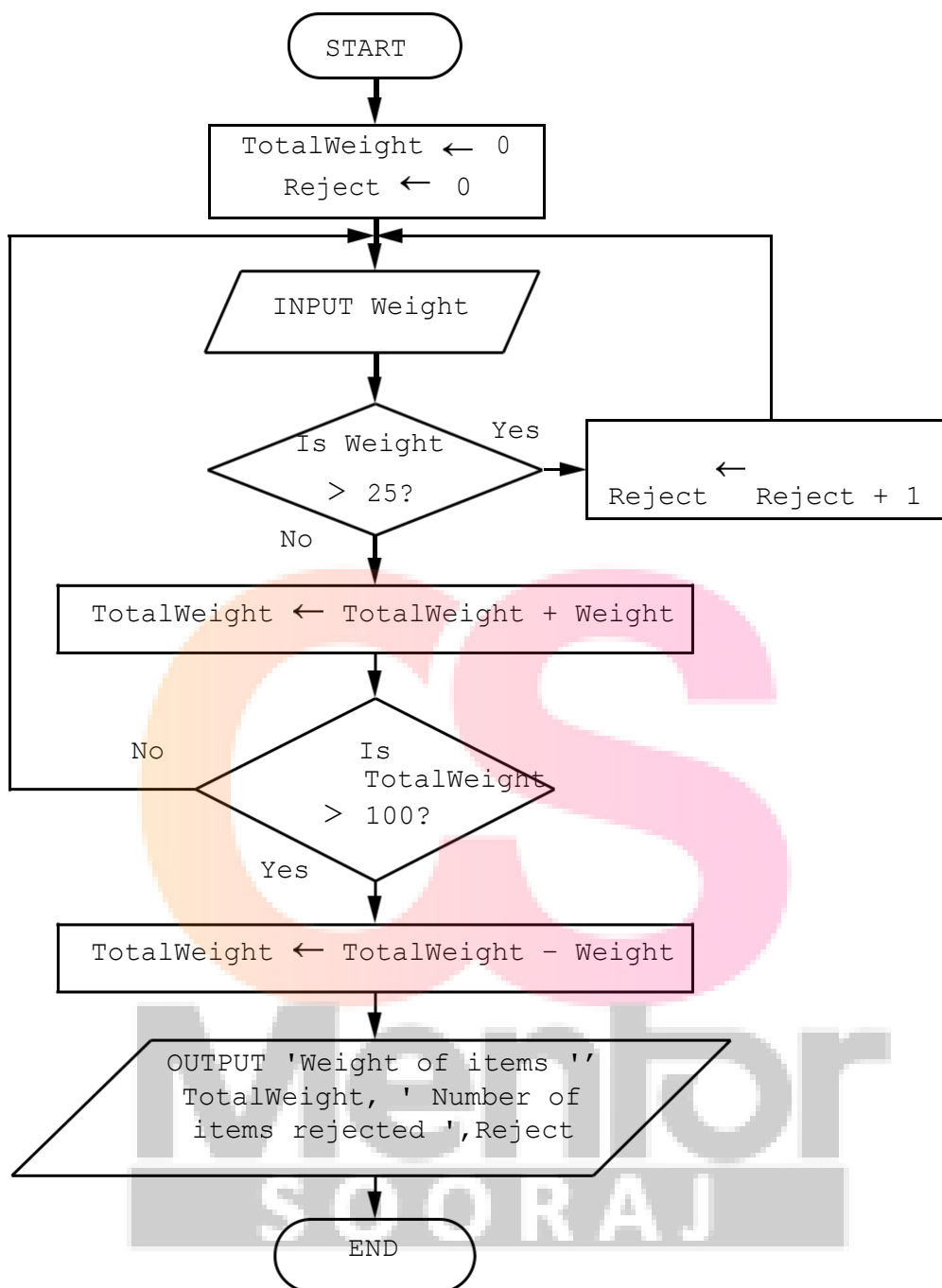
[4]

12 This flowchart inputs the weight of items in kilograms to be loaded on a trailer. Any item over 25 kilograms is rejected. The trailer can take up to 100 kilograms.

```
                        ┌─────────────┐
                        │    START    │
                        └──────┬──────┘
                               ▼
                   ┌───────────────────────┐
                   │ TotalWeight ← 0        │
                   │     Reject ← 0         │
                   └───────────┬───────────┘
                               │
                               ▼
                   ┌───────────────────────┐
                   │    INPUT Weight        │
                   └───────────┬───────────┘
                               ▼
                        ╱╲
                       ╱  ╲        Yes    ┌──────────────────────┐
                      ╱ Is  ╲ ──────────► │          ←           │
                      ╲Weight╱            │  Reject   Reject + 1 │
                       ╲>25?╱             └──────────────────────┘
                        ╲╱
                         │ No
                         ▼
              ┌─────────────────────────────────┐
              │ TotalWeight ← TotalWeight + Weight│
              └─────────────────┬───────────────┘
                                ▼
                   No        ╱╲
              ◄─────────────╱  ╲
                           ╱ Is  ╲
                           ╲Total ╱
                           ╲Weight╱
                            ╲>100?╱
                             ╲╱
                              │ Yes
                              ▼
              ┌─────────────────────────────────┐
              │ TotalWeight ← TotalWeight – Weight│
              └─────────────────┬───────────────┘
                                ▼
              ┌─────────────────────────────────┐
              │ OUTPUT 'Weight of items ''       │
              │  TotalWeight, ' Number of        │
              │  items rejected ',Reject         │
              └─────────────────┬───────────────┘
                                ▼
                        ┌─────────────┐
                        │     END     │
                        └─────────────┘
```

Complete the trace table for the input data:

13, 17, 26, 25, 5, 10, 15, 35, 20, 15

Weight	Reject	TotalWeight	OUTPUT

[5]

13 A programmer has written a routine to check that prices are below $10.00. These values are used as test data.

    10.00   9.99   ten

Explain why each value was chosen.

10.00 ....

....

....

9.99 ....

....

....

ten ....

....

....

                                                                                             [3]

14

The global trade item number (GTIN-8) barcode has seven digits and a check digit.
This pseudocode algorithm inputs seven digits and calculates the eighth digit, then outputs the GTIN-8.
`DIV(X,Y)`, finds the number of divides in division for example `DIV(23,10)` is

2. `MOD(X,Y)`, finds the remainder in division for example `MOD(23,10)` is 3.

```
FOR Count ← 1 TO 7
    INPUT Number
    Digit(Count) ← Number
NEXT
Sum ← (Digit(1)+Digit(3)+Digit(5)+Digit(7))*3+Digit(2)+Digit(4)+Digit(6) IF
MOD(Sum,10) <> 0
    THEN Digit(8) ← DIV(Sum,10)*10 + 10 - Sum
    ELSE Digit(8) ← 0
ENDIF
OUTPUT "GTIN-8"
FOR Count ← 1 TO 8
    OUTPUT Digit(Count)
NEXT
```

(a) Complete the trace table for the input data: 5, 7, 0, 1, 2, 3, 4

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT
5	7	0	1	2	3	4	6	44	GTIN-8
									5 7 0 1 2 3 4 6

Complete the trace table for the input data: 4, 3, 1, 0, 2, 3, 1

Digit(1)	Digit(2)	Digit(3)	Digit(4)	Digit(5)	Digit(6)	Digit(7)	Digit(8)	Sum	OUTPUT
4	3	1	0	2	3	1	0	30	GTIN-8
									4 3 1 0 2 3 1 0

[5]

15 A program checks that the weight of a basket of fruit is over 1.00 kilograms and under 1.10 kilograms. Weights are recorded to an accuracy of two decimal places and any weight not in this form has already been rejected.

Give **three** weights as test data and for each weight state a reason for choosing it. All your reasons must be different.

Weight 1 ....

Reason..

....

Weight 2 ....

Reason..

....

Weight 3 ....

Reason..

....

[3]

16 A programmer has written a routine to store the name, email address and password of a contributor to a website's discussion group.
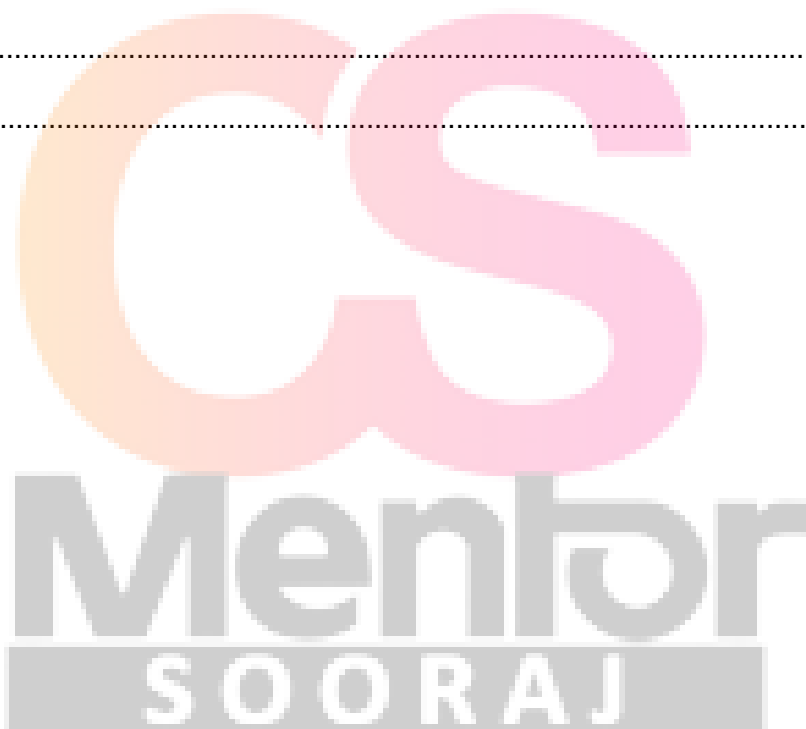
The programmer has chosen to verify the name, email address and password.

Explain why verification was chosen and describe how the programmer would verify this data.

....

....

....

....

....

....

....

....

....[4]

(b) The programmer has also decided to validate the email address and the password.

Describe validation checks that could be used.

Email address ....

....

....

Password ....

....

....
                                                                                                    [2]

17 This pseudocode algorithm inputs two non-zero numbers and a sign, and then performs the calculation shown by the sign. An input of zero for the first number terminates the process.

```
INPUT Number1, Number2, Sign
WHILE Number1 <> 0
   IF Sign = '+' THEN Answer ← Number1 + Number2 ENDIF
   IF Sign = '-' THEN Answer ← Number1 - Number2 ENDIF
   IF Sign = '*' THEN Answer ← Number1 * Number2 ENDIF
   IF Sign = '/' THEN Answer ← Number1 / Number2 ENDIF
   IF Sign <> '/' AND Sign <> '*' AND Sign <> '-' AND Sign <> '+'
      THEN Answer ←  0
   ENDIF
   IF Answer <> 0 THEN OUTPUT Answer ENDIF
   INPUT Number1, Number2, Sign
ENDWHILE
```

(a) Complete the trace table for the input

data: 5,7,+,6,2,–,4,3,$*$,7,8,?,0,0,/

Number1	Number2	Sign	Answer	OUTPUT
5	7	+	12	12
6	2	–	4	4
4	3	*	12	12
7	8	?	0	
0	0	/		

[3]

18 Describe, using an example, the purpose of the following checks during data entry.

Validation check

....

....

....[2]

(b) Verification check

....

....

....[2]
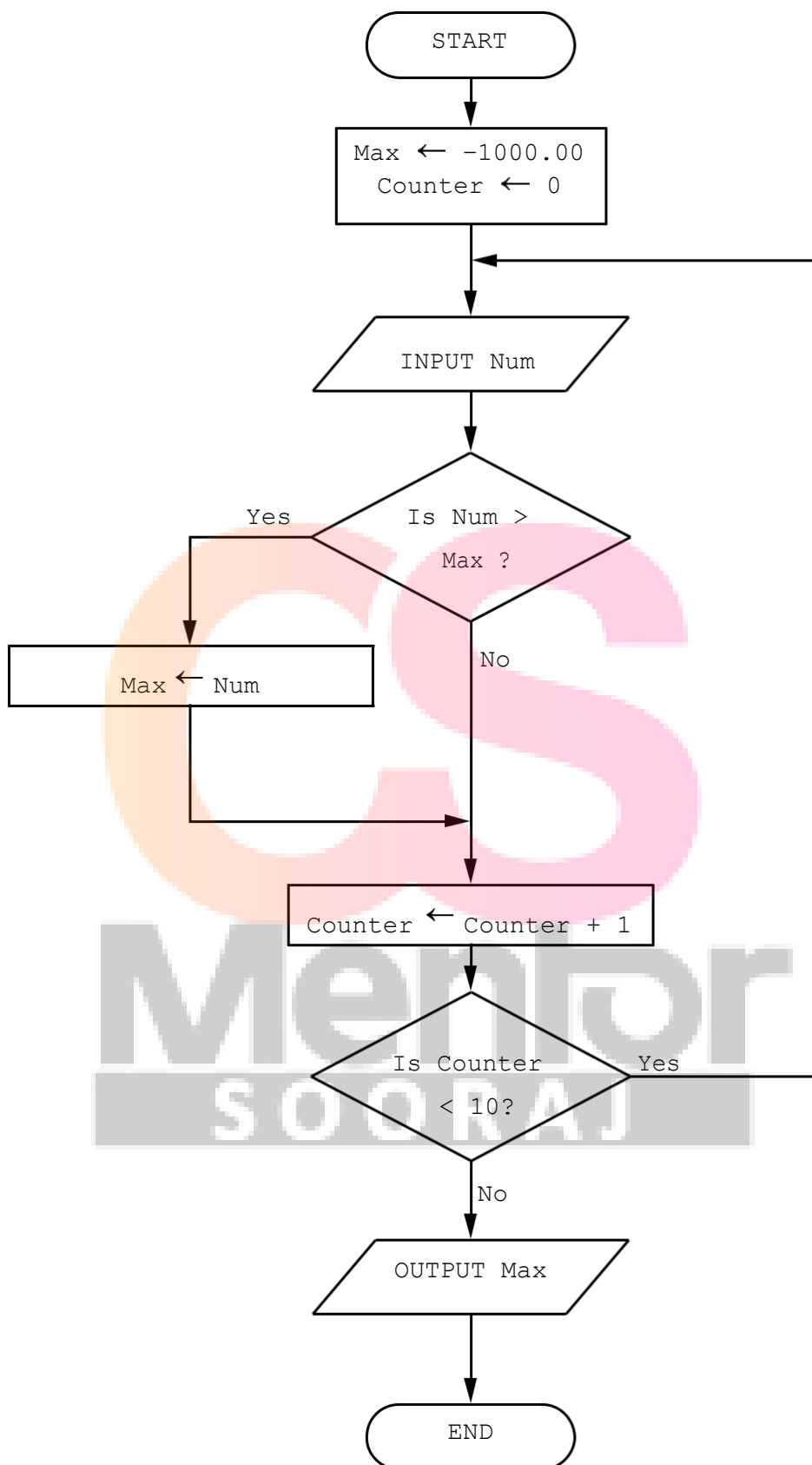
19 An algorithm is written in pseudocode:

```
INPUT Number
IF Number > 100
    THEN OUTPUT "The number is too large"
    ELSE OUTPUT "The number is acceptable"
ENDIF
```

(a) Describe the purpose of the algorithm.

....

....

....

....

....[2]

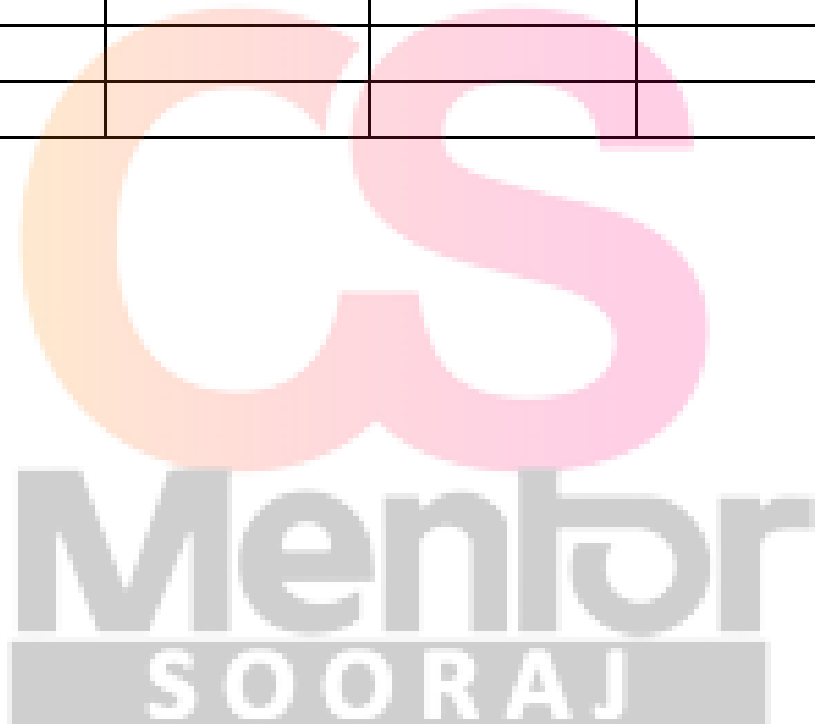**20** The flowchart allows a set of 10 numbers to be entered; it finds and outputs the largest of these numbers.

```
                    ┌─────────────┐
                    │    START    │
                    └──────┬──────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │ Max  ←  −1000.00 │
                  │ Counter  ←  0    │
                  └────────┬─────────┘
                           │
                           ▼
                  ╱──────────────────╲
                  │    INPUT Num     │
                  ╲──────────────────╱
                           │
                           ▼
          Yes      ╱───────────────╲
        ┌──────────│   Is Num >    │
        │          │    Max ?      │
        ▼          ╲───────────────╱
┌──────────────┐          │ No
│ Max  ←  Num  │          │
└──────┬───────┘          │
       │                  │
       └─────────►────────┤
                          ▼
                ┌───────────────────────┐
                │ Counter ← Counter + 1 │
                └───────────┬───────────┘
                            ▼
                   ╱─────────────────╲      Yes
                   │   Is Counter    │──────────►
                   │     < 10?       │
                   ╲─────────────────╱
                            │ No
                            ▼
                  ╱──────────────────╲
                  │   OUTPUT Max     │
                  ╲──────────────────╱
                            │
                            ▼
                    ┌─────────────┐
                    │     END     │
                    └─────────────┘
```

(a) Complete the trace table for the input data:

6.30, 18.62, 50.01, 3.13, 2.05, 50.10, 40.35, 30.69, 0.85, 17.30

Max	Counter	Num	OUTPUT

[3]

21 This section of program code may be used as a validation check.

```
1 PRINT "Input a value between 0 and 100 inclusive"
2 INPUT Value
3   WHILE Value < 0 OR Value > 100
4PRINT "Invalid value, try again"
5INPUT Value
6   ENDWHILE
7   PRINT "Accepted: ", Value
```

Give a name for this type of validation check.

.... [1]

(b) Describe what is happening in this validation check.

....

....

....

....

....

....

....

.... [2]

(c) Complete the trace table for this program code using the test data: 200, 300, −1, 50, 60

Value	OUTPUT

[3]

22 Explain what is meant by **validation** and **verification**.

Give an example for each one.

Validation ....

....

....

....

Example ....

....

....

Verification ....

....

....

....

Example ....

....

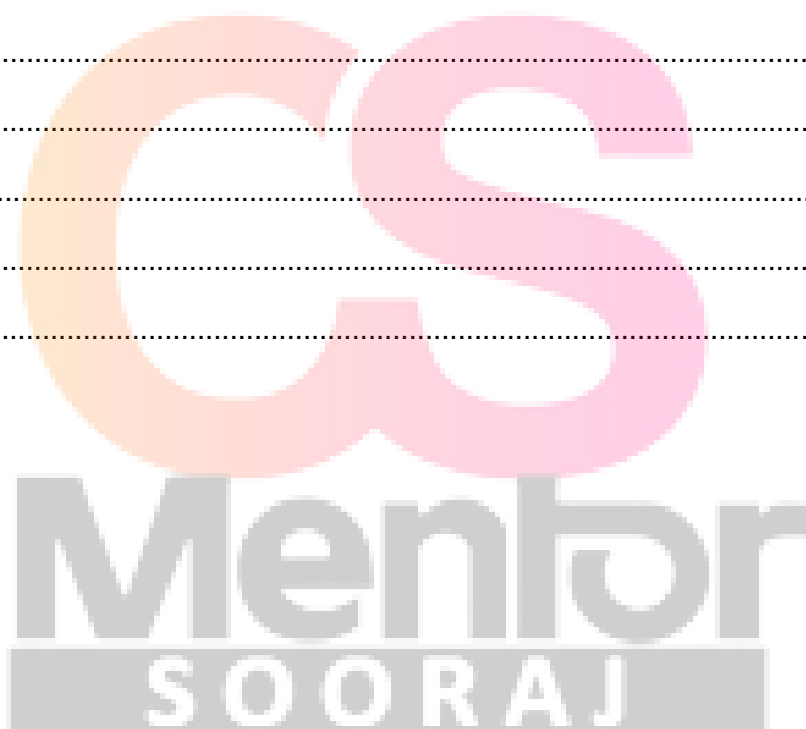...
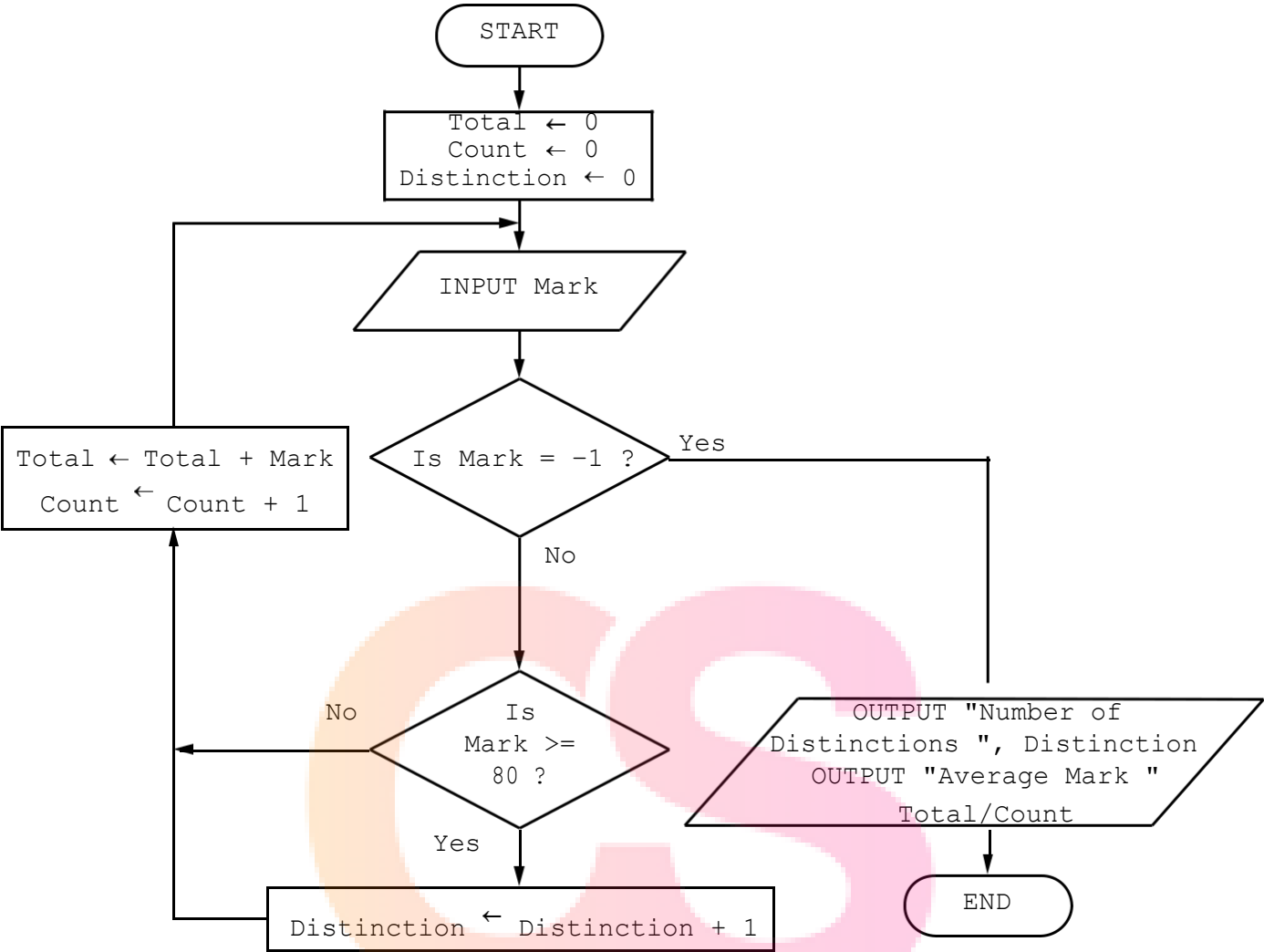
[6]

23 This flowchart inputs the marks gained in an examination. An input of –1 ends the routine.

Complete the trace table for the mark input data: 50, 70, 65, 30, 95, 50, 55, 85, 65, 35, –1, 45

Total	Count	Distinction	Mark	**OUTPUT**
0	0	0		
50	1		50	
120	2		70	
185	3		65	
215	4		30	
310	5	1	95	
360	6		50	
415	7		55	
500	8	2	85	
565	9		65	
600	10		35	
			–1	Number of Distinctions 2 Average Mark 60

[4]

24 (a) An algorithm has been written in pseudocode to input 100 numbers, select and print the largest number and smallest number.

```
Count ← 1
INPUT Number
High ← Number
Low ← Count
REPEAT
  INPUT Number
  IF Number > High
    THEN
       High ← Number
  ENDIF
  IF Number > Low
    THEN
       Low ← Number
  ENDIF
  Count ← Count + 1
UNTIL Count = 99
PRINT "Largest Number is ", Number
PRINT "Smallest Number is ", Low
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1...

Correction ....

....

Error 2...

Correction ....

....

Error 3...

Correction ....

....

Error 4...

Correction ....

....
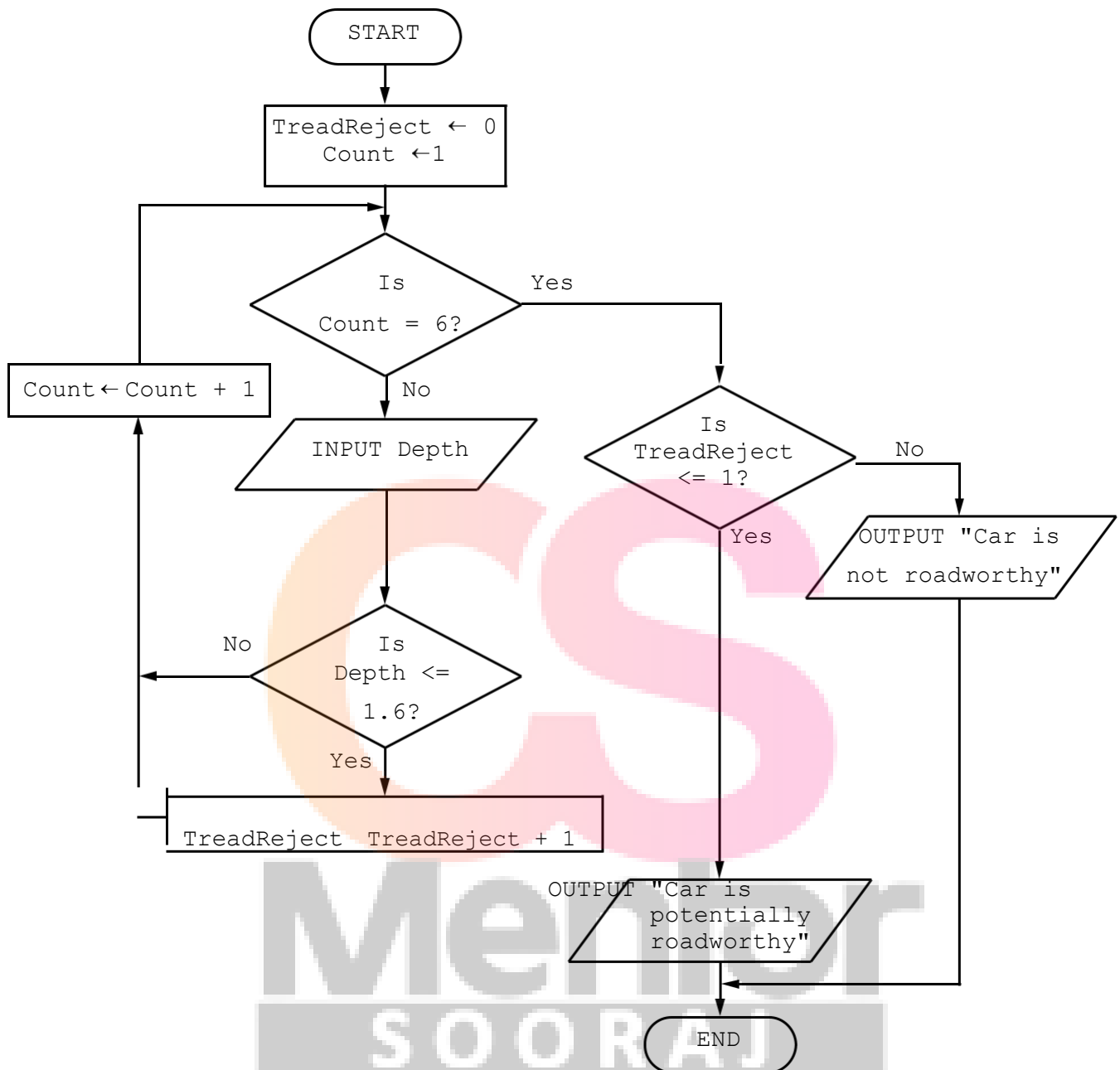
[4]

25 For each of the **four** checks in the table, place a tick in the correct column to show whether it is an example of a **validation** or **verification** check.

Statements	Validation	Verification
Range check		
Double entry		
Check digit		
Presence check		

[4]

26 This flowchart inputs the tread depth of five tyres, four on the car and a spare tyre. Any tread depth of 1.6 mm or less is rejected. To be potentially roadworthy, a car must have four tyres with a tread depth greater than 1.6 mm.

```
                        ( START )
                            │
                            ▼
              ┌─────────────────────────┐
              │ TreadReject ← 0         │
              │   Count ←1              │
              └─────────────────────────┘
                            │
                            ▼
                        ╱ Is ╲            Yes
                     ╱   Count = 6?  ╲─────────────────┐
                        ╲         ╱                    │
                            │ No                       ▼
                            ▼                       ╱  Is  ╲
                    ╱ INPUT Depth ╱              ╱ TreadReject ╲   No
                                             ╲    <= 1?   ╱──────────┐
                                                ╲      ╱             │
                                                   │ Yes            ▼
                            │                                  ╱ OUTPUT "Car is ╱
                            ▼                                  ╱ not roadworthy" ╱
         No             ╱   Is  ╲
      ◄───────────────╱  Depth <=  ╲
                         ╲   1.6?  ╱
                            ╲   ╱
                             │ Yes
                             ▼
              ┌─────────────────────────────┐
              │ TreadReject  TreadReject + 1 │
              └─────────────────────────────┘

                            ╱ OUTPUT "Car is
                            ╱ potentially
                            ╱ roadworthy"
                                  │
                                  ▼
                              ( END )
```

Complete Trace table 1 for the tread depth input data:
1.7, 1.9, 1.4, 1.8, 2.0

TreadReject	Count	Depth	OUTPUT

Trace table 1

Complete Trace table 2 for the tread depth input data:
1.2, 1.9, 1.4, 1.8, 2.4

TreadReject	Count	Depth	OUTPUT

Trace table 2

[4]

27 (a) An algorithm has been written in pseudocode to input the weight of 500 items and reject any that are over-weight or under-weight, then print the percentage rejected.

```
Count ← 1
Reject ← 0
Over ← 62
Under ← 58
REPEAT
  INPUT ItemWeight
  IF ItemWeight > Over AND ItemWeight < Under
    THEN
      Reject ← Reject - 1
  ENDIF
  Count ← Count + 1
UNTIL Count > = 500
Reject ← Reject / 100
PRINT "Percentage rejected is ", Reject
```

Find the **four** errors in the pseudocode and suggest a correction for each error.

Error 1 ....

....

Correction ....

....

Error 2 ....

....

Correction ....

....

Error 3 ....

....

Correction ....

....

Error 4 ....

....

Correction ....

....

[4]

28 Tick (✔) **one** box in each row to identify if the statement about structure diagrams is true or false.

Statement	True (✔)	False (✔)
A structure diagram is a piece of code that is available throughout the structure of a program.		
A structure diagram shows the hierarchy of a system.		
A structure diagram is another name for an array.		
A structure diagram shows the relationship between different components of a system.		

[2]

29 Programs can perform validation and verification checks when data is entered.

Give the names of **two** different validation checks and state the purpose of each one. Check

1 ....

Purpose ....

....

....

Check 2 ....

Purpose ....

....

....

[4]

(b) Give the name of **one** verification check.

.... [1]

(c) Describe the difference between validation and verification.

....

....

....

.... [2]
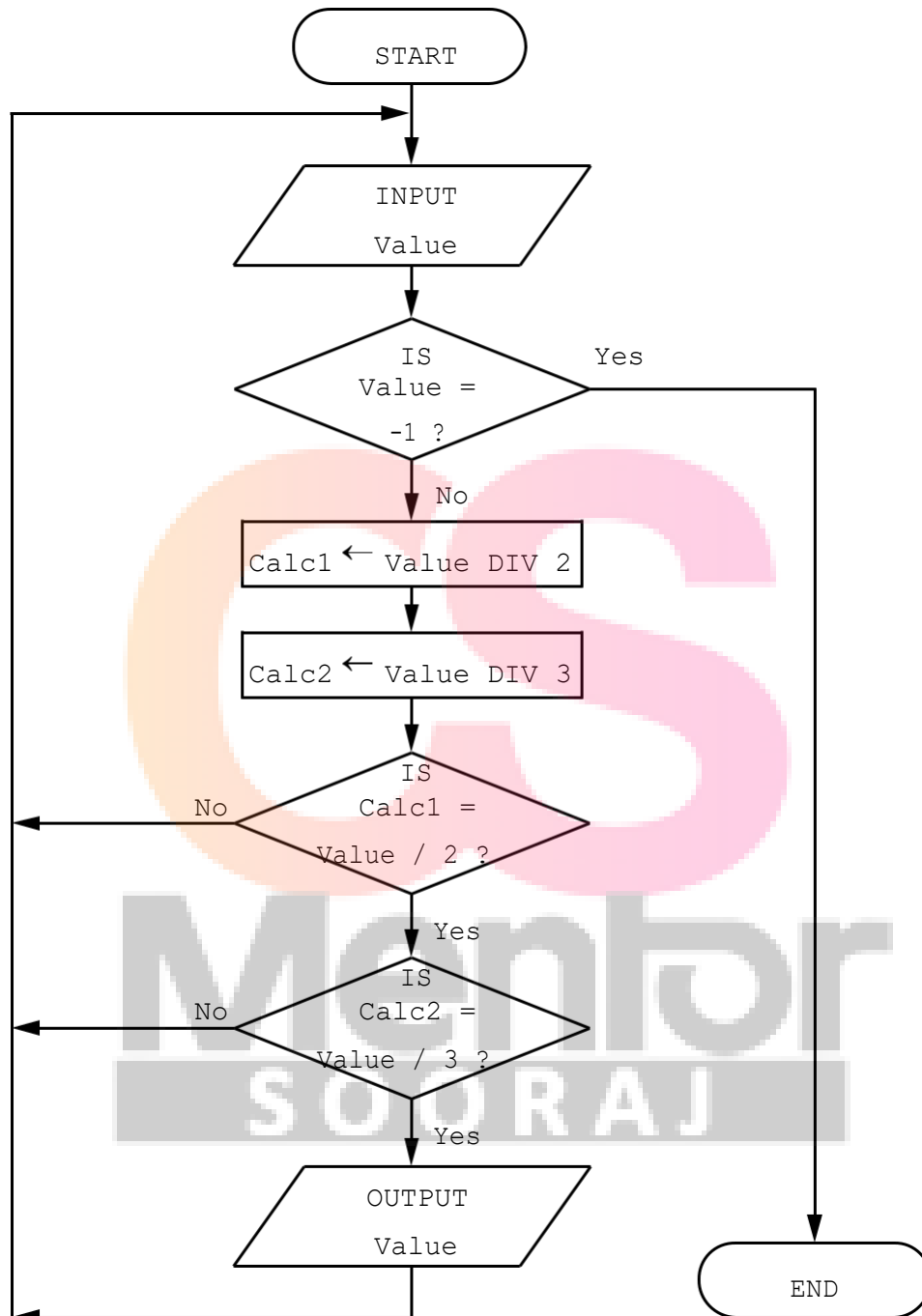
30 The flowchart represents an algorithm.
The predefined function DIV gives the value of the result of integer division,
for example, $y \leftarrow 9$ DIV 4 gives y a value of 2

An input value of −1 ends the algorithm.

```
                          ┌──────────────┐
                          │    START     │
                          └──────┬───────┘
                                 │
   ┌──────────────────────────── │
   │                             ▼
   │                     ╱─────────────────╲
   │                    ╱       INPUT        ╲
   │                    ╲       Value        ╱
   │                     ╲─────────────────╱
   │                             │
   │                             ▼
   │                        ╱─────────╲
   │                       ╱    IS     ╲        Yes
   │                      ╱   Value =    ╲──────────────┐
   │                      ╲    -1 ?     ╱               │
   │                       ╲─────────╱                  │
   │                            │ No                    │
   │                            ▼                        │
   │                 ┌────────────────────────┐          │
   │                 │ Calc1 ← Value DIV 2     │          │
   │                 └───────────┬────────────┘          │
   │                             ▼                        │
   │                 ┌────────────────────────┐          │
   │                 │ Calc2 ← Value DIV 3     │          │
   │                 └───────────┬────────────┘          │
   │                             ▼                        │
   │           No         ╱─────────────╲                 │
   │ ◄───────────────────╱      IS       ╲                │
   │                     ╲    Calc1 =     ╱               │
   │                      ╲ Value / 2 ?  ╱                │
   │                       ╲───────────╱                  │
   │                             │ Yes                    │
   │                             ▼                        │
   │           No         ╱─────────────╲                 │
   │ ◄───────────────────╱      IS       ╲                │
   │                     ╲    Calc2 =     ╱               │
   │                      ╲ Value / 3 ?  ╱                │
   │                       ╲───────────╱                  │
   │                             │ Yes                    │
   │                             ▼                        ▼
   │                    ╱─────────────────╲      ┌──────────────┐
   │                   ╱     OUTPUT        ╲     │     END      │
   │                   ╲     Value         ╱     └──────────────┘
   │                    ╲─────────────────╱
   └─────────────────────────────┘
```

(a) Complete the trace table for the input

data: 50, 33, 18, 15, 30, −1, 45, 12, 90, 6

Value	Calc1	Calc2	OUTPUT

[4]

(b) Describe the purpose of the algorithm.

....

....

.... [2]
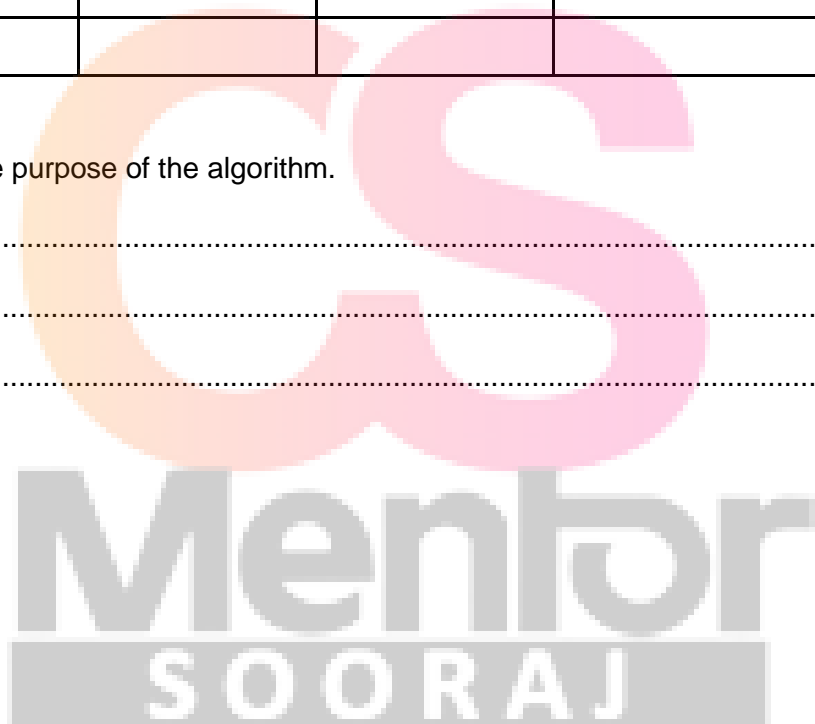
31 This flowchart inputs the points won and the points lost when playing a game. The difference between the points won and lost is calculated and depending on the result the player can: move up to the next level, stay at the same level, or move down to the previous level. The flowchart finishes when the input for points won is –1.
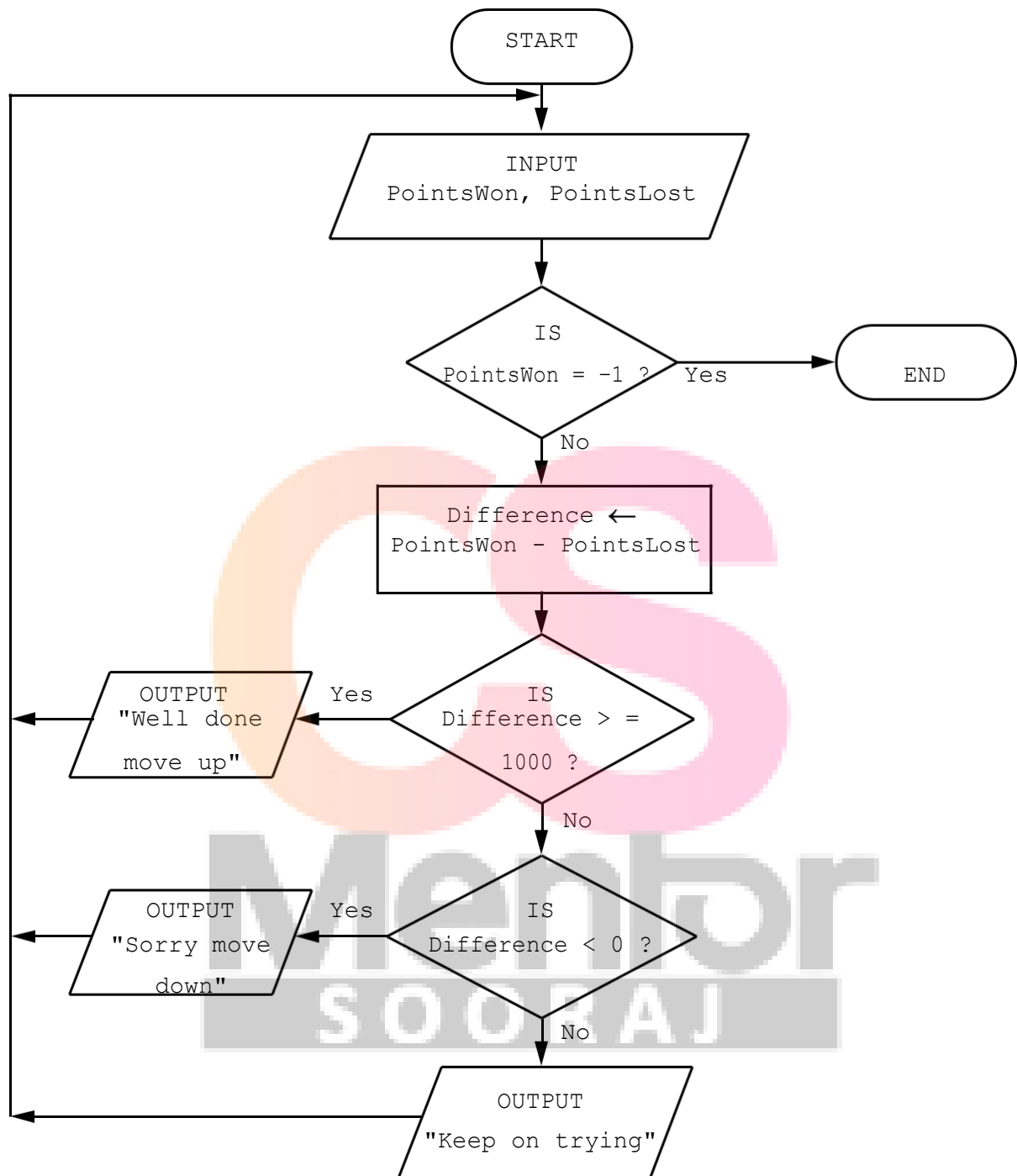
(a) Complete a trace table for this set of input data:
5000, 4474, 6055, 2000, 7900, 9800, 3000, 2150, –1, 6700, 7615

PointsWon	PointsLost	Difference	OUTPUT

[3]

32 (a) An algorithm has been written in pseudocode to input the names and marks of 35 students. The algorithm stores the names and marks in two arrays `Name[ ]` and `Mark[ ]`. The highest mark awarded is found and the number of students with that mark is counted. Both of these values are output.

```
1    HighestMark ← 100  ←
2    HighestMarkStudents ← 0
3    FOR Count1 TO 35
4        OUTPUT "Please enter student name"
5        INPUT Name[Count]
6        OUTPUT "Please enter student mark"
7        INPUT Mark[Counter]
8        IF Mark[Count] = HighestMark
09         THEN
10            HighestMarkStudents ← HighestMarkStudents – 1
11         ENDIF
12         IF Mark[Count] > HighestMark
13           THEN
14              Mark[Count] ← HighestMark
15              HighestMarkStudents ← 1
16         ENDIF
17   NEXT Count
18   OUTPUT "There are ", HighestMarkStudents," with the highest mark of ",
        HighestMark
```

Give line numbers where the **four** errors are to be found in the pseudocode. Suggest a correction for each error.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

Error 4 line number ....

Correction ....

....

[4]

33 A code must take the form LL9 9LL where L is a letter and 9 is a digit.

A presence check has already been used to ensure data has been entered. Name **two** other types of validation check that can be used to test the code is valid.

Check 1 ....

Check 2 ....

[2]

(b) Give **one** example of invalid test data for each of the validation checks you have named in **part (a)** and in each case, give a reason why it fails the check. Each example of test data must be different.

Check 1 Invalid Test Data ....

....

Reason ....

....

....

....

Check 2 Invalid Test Data ....

....

Reason ....

....

....

....
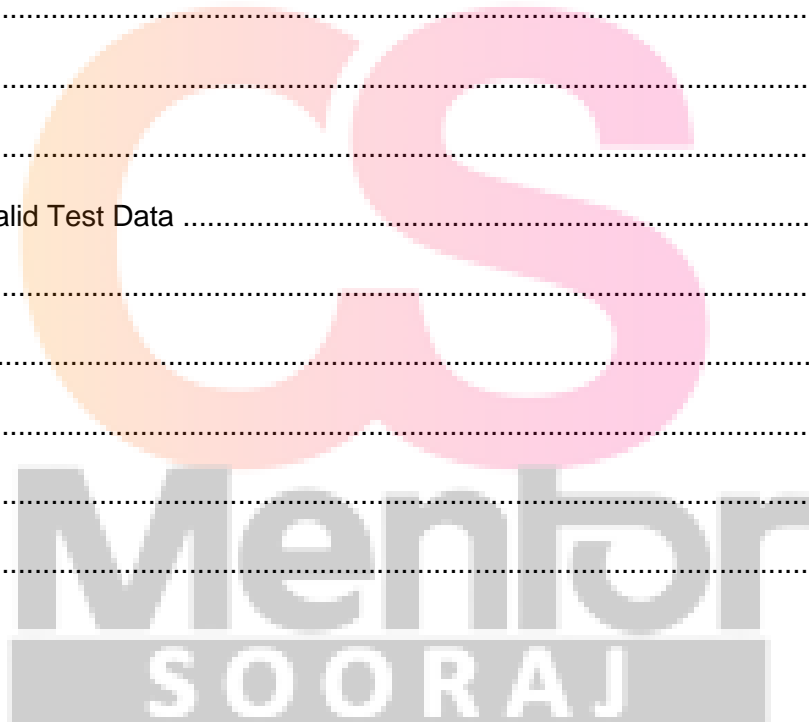
[4]

34 This algorithm finds prime numbers.

The pre-defined function DIV gives the value of the result of integer division,

for example, $y \leftarrow 9\ \text{DIV}\ 4$ gives y a value of 2

```
Flag ← False
INPUT Number
WHILE Number <> 0
  Divisor ← 2
  WHILE Divisor <= Number / 2
    Value ← Number DIV Divisor
    IF Number / Divisor = Value
      THEN
        Flag ← True
    ENDIF
    Divisor ← Divisor + 1
  ENDWHILE
  IF Flag = False
    THEN
      OUTPUT Number, " is prime"
  ENDIF
INPUT Number
Flag ← False
ENDWHILE
```

Complete the trace table for the algorithm using the input data:

5, 6, 8, 0, 11, 13

Flag	Number	Divisor	Value	OUTPUT
False	5	2	2	
		3		5 is prime
False	6	2	3	
True		3	2	
		4		
False	8	2	4	
True		3	2	
		4	2	
True		5		
	0			

[5]

35 This section of pseudocode is to be used as a validation check that will continue until a number between 0 and 499 inclusive is entered.

```
1       PRINT "Input a number from 0 to 499 inclusive"
2       FOR Number ← 1 TO 10
3         INPUT Number
4         IF Number < 0 AND Number > 499
5           THEN
6             PRINT "Invalid number, please try again"
7           ENDIF
8       UNTIL Number = 0 OR Number = 499
9       PRINT Number, " is within the correct range"
```

There are **three** lines in this pseudocode that contain errors. In each case, state the line number to identify the incorrect line and write out the corrected line in full.

Error 1 line number ....

Correction ....

....

Error 2 line number ....

Correction ....

....

Error 3 line number ....

Correction ....

....

[6]