7. Algorithm design and problem-solving

1 Four validation checks and four descriptions are shown below.

Draw a line to link each validation check to the correct description.

Validation check	Description
Presence check	Numbers between two given values are accepted
Range check	Data is of a particular specified type
Type check	Data contains an exact number of characters
Length check	Ensures that some data have been entered

[3]

2 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]

3 **(a)** Explain the difference between a validation check and a verification check.

.....

.....

.....[2]

(b) Describe, using an example, how data could be verified on data entry.

.....

.....

.....[2]

(c) Explain what is meant by the term library routine.

.....

.....

.....[2]

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

(a) 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]

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

(a) Validation check

.....

.....

.....[2]

(b) Verification check

.....

.....

.....[2]

6 Describe, giving a different example for each, the purpose of these validation checks used in programming.

Range check .....

.....

.....

Example .....

.....

.....

Length check .....

.....

.....

Example .....

.....

.....

Type check .....

.....

.....

Example .....

.....

.....

[6]

7 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

.....

.....

.....

.....

..... [2]

(b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

.....

.....

.....

..... [2]

(d) Draw a flowchart to represent this section of program code.

8 Explain what is meant by **validation** and **verification**.
Give an example for each one.

Validation .....

.....

.....

.....

Example .....

.....

.....

Verification .....

.....

.....

.....

Example .....

.....

.....

[6]

9 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]

10 **Four** validation checks and **four** descriptions are shown.

Draw a line to connect each validation check to the correct description.

Validation Check	Description
Range check	Checks that some data is entered.
Presence check	Checks for a maximum number of characters in the data entered.
Length check	Checks that the characters entered are all numbers.
Type check	Checks that the value entered is between an upper value and a lower value.

[3]

11 Describe the use of a subroutine in a program.

.....

.....

.....

.....

..... [2]

12 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]

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

(a) 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]

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

(a) 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]

15 Tick (✓) **one** box in each row to identify if the statement about subroutines is **true** or **false**.

Statement	true (✓)	false (✓)
A subroutine is called from within a program.		
A subroutine is **not** a complete program.		
A subroutine is a self-contained piece of code.		
A subroutine must return a value to the code from which it was called.		

[2]

16 Four programming concepts and five descriptions are shown.

Draw a line to connect each **Programming concept** to its correct **Description**. Not all Descriptions will be connected to a Programming concept.

Programming concept	Description
Validation	A subroutine that does not have to return a value
Verification	An automatic check to ensure that data input is reasonable and sensible
Procedure	A subroutine that always returns a value
Function	An overview of a program or subroutine
	A check to ensure that data input matches the original

[4]

17 Tick (✓) **one** box in each row to identify if the statement is about **validation**, **verification** or **both**.

Statement	Validation (✓)	Verification (✓)	Both (✓)
Entering the data twice to check if both entries are the same.			
Automatically checking that only numeric data has been entered.			
Checking data entered into a computer system before it is stored or processed.			
Visually checking that no errors have been introduced during data entry.			

[3]

18 A program has been written to check the value of a measurement. The measurement must be a positive number and given to three decimal places, for example, `3.982`

(a) (i) State suitable examples of normal and erroneous test data that could be used to test this program. For each example give the reason for your choice of test data.

Normal test data example .....

Reason .....

.....

Erroneous test data example .....

Reason .....

.....

[4]

(ii) Explain why two pieces of boundary test data are required for this program.
Give an example of each piece of boundary test data.

.....

.....

.....

.....

..... [3]

(b) Explain why verification is needed and how verification could be performed by this program.

.....

.....

.....

.....

.....

..... [3]

19 Tick (✓) one box in each row to identify if the statement is about validation, verification or neither.

Statement	Validation (✓)	Verification (✓)	Neither (✓)
a check where data is re-entered to make sure no errors have been introduced during data entry			
an automatic check to make sure the data entered has the correct number of characters			
a check to make sure the data entered is sensible			
a check to make sure the data entered is correct			

[3]

20 A program checks that the data entered is between 1 and 100 inclusive.

Identify **one** piece of normal, extreme and erroneous test data for this program, and give a reason for each.

Normal test data .....

Reason .....

.....

.....

Extreme test data .....

Reason .....

.....

.....

Erroneous test data .....

Reason .....

.....

.....

[6]

21 Give **one** piece of normal test data and **one** piece of erroneous test data that could be used to validate the input of an email address.

State the reason for your choice in each case.

Normal test data .....

.....

Reason .....

.....

.....

.....

Erroneous test data .....

.....

Reason .....

.....

.....

.....

[4]

22 Tick (✓) **one** or more boxes in each row to match the type(s) of test data to each description.

Description	Types of test data			
	Boundary	**Erroneous / Abnormal**	**Extreme**	**Normal**
test data that is always on the limit of acceptability				
test data that is either on the limit of acceptability or test data that is just outside the limit of acceptability				
test data that will always be rejected				
test data that is within the limits of acceptability				

[4]

23 **(a)** A PIN (personal identification number) is input into a banking app by the user. Before the PIN is accepted, the following validation checks are performed:

- check 1 – each character must be a digit
- check 2 – there must be exactly four digits
- check 3 – the value of the PIN must be between 1000 and 9999 inclusive.

Describe each validation check.

Check 1 .....

.....

.....

.....

Check 2 .....

.....

.....

.....

Check 3 .....

.....

.....

.....

[6]

(b) The PIN can be changed by the user.

Describe how the new PIN could be verified before use.

.....

.....

.....

.....

.....

..... [3]

24 Draw a line to connect each programming concept to the most appropriate description.

Programming concept	Description
counting	carrying out an action multiple times within a loop structure
repetition	adding together the numbers in a list of numbers
selection	tracking the number of iterations a program has performed in a loop
sequence	branching off to take a course of action depending on the answer to a question
totalling	a set of statements to be executed in order

[4]

25 Describe the use of verification on input of data when entering a list of items in stock into a database. Explain why verification is necessary.

.....

.....

.....

.....

.....

..... [3]

26 Describe **one** type of test data that must be used to test if a program accepts valid input data.

.....

.....

.....

..... [2]

27 A check digit is to be used to validate an identification number on input. The identification number contains five digits and the check digit.
The check digit is calculated by adding up the first five digits, dividing by 10 and taking the remainder.
For example, `5 + 1 + 2 + 4 + 3` divided by `10` gives a remainder of `5` so the six-digit identification number would be `512435`

(a) (i) Calculate the check digit for `69321`

.....

..... [1]

Working space .....

.....

.....

(ii) State which of these identification numbers have incorrect check digits.

A `123455`
B `691400`
C `722855`
D `231200`

.....

..... [2]

Working space .....

.....

.....

(b) **(i)** Describe an input error that would **not** be found using this check digit.

.....

.....

.....

..... [2]

(ii) Describe a more suitable algorithm to calculate the check digit for this identification number.

.....

.....

.....

..... [2]

(c) Identify **two** other validation checks that could be used when inputting this identification number.

1 .....

.....

2 .....

.....

[2]

28 **(a)** **Four** descriptions of stages in the program development life cycle are shown.

Draw **one** line to link each description to its most appropriate program development life cycle stage.

Not all program development life cycle stages will be used.

Program development life cycle description	Program development life cycle stage
develop an algorithm to solve the problem by using structure diagrams, flowcharts or pseudocode	analysis
detect and fix the errors in the program	coding
identify the problem and its requirements	design
write and implement the instructions to solve the problem	evaluation
	testing

[4]

(b) Identify **three** of the component parts after a problem has been decomposed.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

29 Tick (✓) **one** box to show the name of the data structure used to store a collection of data of the same data type.

A Array ☐

B Constant ☐

C Function ☐

D Variable ☐

[1]

30 **(a)** Describe what is meant by data validation.

.....

.....

.....

..... [2]

(b) A validation check is used to make sure that any value that is input is an integer between 30 and 200 inclusive.

Give **one** example of each type of test data to check that the validation check is working as intended. Each example of test data must be different.

Give a reason for each of your choices of test data.

Normal test data .....

Reason .....

.....

Abnormal test data .....

Reason .....

.....

Extreme test data .....

Reason .....

.....

[6]

31 Tick (✓) **one** box to identify the first stage of the program development life cycle.

A Analysis ☐

B Coding ☐

C Design ☐

D Testing ☐

[1]

32 Identify **three** different ways that the design of a solution to a problem can be presented.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

33 A program needs to make sure the value input for a measurement meets the following rules:

- the value is a positive number
- a value is always input
- the value is less than 1000.

(a) Describe the validation checks that the programmer would need to use.

.....

.....

.....

.....

.....

..... [3]

(b) The program needs editing to include a double entry check for the value input.

(i) State why this check needs to be included.

.....

..... [1]

(ii) The input value needs to be stored in the variable `Measurement`
Write pseudocode to perform the double entry check until a successful input is made.

.....

.....

34 **(a)** Explain why verification checks are used when data is input.

.....

.....

.....

..... [2]

(b) Give **two** types of verification check and state how each one can be used.

Verification check 1 .....

Use .....

.....

Verification check 2 .....

Use .....

.....

[4]

35 **(a)** **Four** descriptions of validation checks are shown.

Draw **one** line to link each description to the most appropriate check.

Not all checks will be used.

Description	Check
to check that the data entered is an integer	check digit
to check that some data has been entered	format check
to check that the data entered has an appropriate number of characters	length check
to check that an identification number contains no errors	presence check
	type check

[4]

(b) Write an algorithm in pseudocode to make sure that an input for the variable `Length` is between 15 and 35 inclusive. The code must iterate until a valid input has been made and the code must include appropriate messages.

.....

.....

.....

.....

.....

.....

.....

..... [3]

36 Circle the **three** words representing places where data may be stored.

array constant dimension input

output procedure variable

[3]

37 The first stage of the program development life cycle is analysis. Two of the tasks in analysis are abstraction and decomposition.

(a) Describe what is meant by abstraction.

.....

.....

.....

..... [2]

(b) Identify **three** of the component parts when a problem has been decomposed at the analysis stage.

1 .....

2 .....

3 .....

[3]

(c) Identify and describe **one** other stage of the program development life cycle.

.....

.....

.....

..... [2]

38 A programmer is designing an algorithm to calculate the cost of a length of rope.
The program requirements are:

- input two values: the length of rope in metres `Length` and the cost of one metre `Cost`
- perform a validation check on the length to ensure that the value is between `0.5` and `6.0` inclusive
- calculate the price `Price`
- output the price rounded to two decimal places.

Use the variable names given.

(a) State the name of the validation check.

..... [1]

(b) Complete the flowchart for this algorithm.

[6]

(c) Give **two** different sets of test data for this algorithm and state the purpose of each set.

Set 1 .....

Purpose .....

.....

.....

Set 2 .....

Purpose .....

.....

.....

[4]

(d) Complete the headings for the trace table to show a dry-run for this algorithm.
You do **not** need to trace the algorithm.

.....	.....	.....	.....

[3]

(e) Describe an improvement that should be made to the requirements for this algorithm.

.....

.....

.....

[2]

39 Tick (✓) **one** box to complete the sentence.

Verification is used to make sure that a value entered

A has **not** changed during input. ☐

B is an integer. ☐

C is correct. ☐

D is **not** a string. ☐

[1]

40 A type of validation check is a length check. Another type of validation check is used to make sure that any date entered is in the dd/mm/yyyy style:
dd means day, mm means month and yyyy means year.

(a) State the type of validation check used.

..... [1]

(b) Give **one** example of normal test data and **one** example of abnormal test data you should use to make sure the check **in part (a)** is working properly.

State a reason for each of your choices of test data.

Normal .....

Reason .....

.....

Abnormal .....

Reason .....

.....

[4]

(c) Describe how a length check could be used with the date entered.

.....

.....

.....

..... [2]

41 Tick (✓) **one** box to show which term is an example of a verification check.

A Double entry check ☐

B Format check ☐

C Length check ☐

D Presence check ☐

[1]

42 A programmer is writing a data entry program for booking theatre seats.
The programmer needs the program to accept only whole numbers that are greater than or equal to one and less than or equal to six.

(a) Give the names of **two** validation checks that are required for this program.

1 .....

2 .....

[2]

(b) Complete this pseudocode to perform your **two** validation checks, using your answers given in **(a)**:

```
OUTPUT "Please enter the number of seats you want to book "
INPUT Seats
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

[5]

(c) Give **one** item of test data to use when testing this program.
State the reason for your choice of test data.

Test data .....

Reason for choice .....

.....

[2]

43 Tick (✓) **one** box to complete this sentence.

A validation check to make sure that an email address contains an '@' sign is a

A range check. ☐

B visual check. ☐

C presence check. ☐

D format check. ☐

[1]

44 Tick (✓) **one** box to identify a method used to design and construct a solution to a computing problem.

A analysis ☐

B coding ☐

C flowchart ☐

D testing ☐

[1]

45 Identify **three** different tasks in the analysis stage of the program development life cycle.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

46 A program is to be written that will accept integers that are between the values of 1 and 80 inclusive.

(a) Give examples of normal, abnormal and extreme test data that could be used to make sure the program only accepts valid data.

Normal test data .....

Abnormal test data .....

Extreme test data .....

[3]

(b) Describe what is meant by extreme test data.

.....

.....

.....

..... [2]

47 Tick (✓) **one** box to show which check is used for verification when data is input.

A length check ☐

B range check ☐

C type check ☐

D visual check ☐

[1]

48 Identify **three** stages of the program development life cycle from the following list of words.

analysis **decomposition** **design** **input**

pseudocode **testing** **variable**

1 .....

2 .....

3 .....

[3]

49 Describe **three** methods that are used to design and construct a solution to a problem.

Method 1 .....

.....

.....

.....

Method 2 .....

.....

.....

.....

Method 3 .....

.....

.....

.....

[6]

50 Tick (✓) **one** box to complete this sentence.

A solution to a problem may be represented using pseudocode, flowcharts or

A procedures. ☐

B processes. ☐

C structure diagrams. ☐

D sub-systems. ☐

[1]

51 Tick (✓) **one** box to complete this sentence.

A pseudocode example of a selection statement is

A `CALL Sorting(Value1, Value2)` ☐

B `DECLARE Count : INTEGER` ☐

C `IF X = 7 THEN Y ← 21 ENDIF` ☐

D `WHILE X <> -1 DO` ☐

[1]

52 Analysis is one stage in the program development life cycle.

(a) State **one** other stage in the program development life cycle.

..... [1]

(b) Describe the analysis stage of the program development life cycle.

.....

.....

.....

.....

.....

..... [3]

53 Outline **one** type of verification check that could be used when inputting data.

.....

.....

.....

..... [2]

54 Different types of test data are used during program development to make sure a program works as intended. A program being developed takes as input whole numbers that are **not** greater than 80.

Identify **two** items of test data to test the whole number limit of 80.

Explain the reason for your choice of the data in each case.

Test data 1 .....

Reason for choice .....

.....

Test data 2 .....

Reason for choice .....

.....

[4]

55 Tick (✓) **one** box to show which of the following is used to validate data on input.

A checksum ☐

B double entry check ☐

C type check ☐

D visual check ☐

[1]

56 Tick (✓) **one** box to show a method used to construct a solution to a problem.

A abstraction ☐

B structure diagram ☐

C test data ☐

D variable ☐

[1]

57 One stage of the program development life cycle is the analysis stage.

Identify and describe **two** other stages of the program development life cycle.

Stage .....

Description .....

.....

.....

.....

.....

Stage .....

Description .....

.....

.....

.....

.....

[6]

Pseudocode

58 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]

59 Explain the difference between a variable and a constant in a program.

.....

.....

.....

.....[2]

60 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]

61 Four programming concepts and four examples of programming code are shown below.

Draw a line to link each programming concept to the correct example of programming code.

Programming concept	Example of programming code
Counting	`Sum = Sum + Value[n]`
Repetition	`IF Value = 10 THEN PRINT 'X'`
Selection	`FOR Counter = 1 TO 10`
Totalling	`Amount = Amount + 1`
	`Sum = Num1 + Num2`

[4]

62 Read this section of program code that should input 50 numbers and then output the average.

```
1   Total = 0
2   For Counter = 1 TO 50
3     INPUT Num
4     Total = Total + 1
5     Counter = Counter + 1
6     Average = Total/Counter
7   NEXT Counter
8   PRINT Average
```

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

Locate these errors and suggest code corrections to remove each error.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....[4]

63 A routine checks the weight of melons to be sold in a supermarket. Melons weighing under 0.5 kilograms are rejected and melons weighing over 2 kilograms are also rejected.

Give an example of each type of test data for this routine.

Normal .....

Extreme .....

Abnormal .....[3]

64 Identify **two** different conditional statements that you can use when writing pseudocode.

1 .....

2 .....[2]

65 Read this section of program code that should input 50 numbers and then output the average of the positive numbers only.

```
1   Total = 0
2   PosCount = 0
3   FOR Counter = 1 TO 50
4     INPUT Num
5     IF Num < 0 THEN Total = Total + Num
6     IF Num > 0 THEN Counter = Counter + 1
7     Average = Total/PosCount
8   NEXT Counter
9   PRINT Num
```

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

Locate these errors and suggest code corrections to remove each error.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....[4]

66 A routine checks the age and height of children who are allowed to enter a play area. The children must be less than 5 years of age and under 1 metre in height.

(a) The first set of test data used is age 3 and height 0.82 metres.

State what type of test data this is.

.....

Give a reason for using this test data.

.....

.....[2]

(b) Provide **two** additional sets of test data. For each, give

- the type of each set of test data
- the reason why it is used

Each type of test data and reason for use must be different.

Set 1 .....

Type .....

Reason .....

.....

.....

Set 2 .....

Type .....

Reason .....

.....

[6]

67 Read this section of program code that inputs 10 positive numbers and then outputs the smallest number input.

```
1   Small = 1000
2   Counter = 0
3   REPEAT
4     INPUT Num
5     IF Num < Small THEN Small = Num
6     Counter = Counter + 1
7   UNTIL Counter = 10
8   PRINT Small
```

(i) Identify **three** changes you would need to make to find the largest number input instead of the smallest number.

1 .....

.....

2 .....

.....

3 .....

.....[3]

(ii) Rewrite the program code with your changes.

.....

.....

.....

.....

.....

.....

.....

.....[3]

68 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.

(i) Suggest **three** improvements that could be made.

1.....

.....

2.....

.....

3.....

.....[3]

(ii) Rewrite the program code with your improvements.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

69 Read this section of program code that:

- inputs 10 numbers
- checks whether each number is within a specified range
- totals the numbers within the range and outside the range

```
1  InRange = 0
2  OutRange = 1000
3  FOR Count = 1 TO 10
4    INPUT Num
5    IF Num > 10 AND Num < 20 THEN InRange = InRange + 1
6      ELSE OutRange = OutRange - 1
7  Count = Count + 1
8  NEXT X
9  PRINT InRange, OutRange
```

(a) There are four errors in this code.

Locate these errors and suggest a correction to remove each error.

Error 1.....

Correction .....

.....

Error 2.....

Correction .....

.....

Error 3.....

Correction .....

.....

Error 4.....

Correction .....

.....[4]

(b) Decide, with reasons, whether the numbers 10 and 20 are within or outside the range.

Number	Within range (✓)	Outside range (✓)	Reason
10			....
20			....

[4]

70 Read this section of program code that inputs positive numbers, discards any negative numbers and then outputs the average. An input of zero ends the process.

```
1  Total = 0
2  Counter = 100
3  REPEAT
4    REPEAT
5      INPUT Num
6    UNTIL Num < 0
7    Total = Total + 1
8    Counter = Counter + Num
9  UNTIL Num = 0
10 Average = Total / (Counter - 1)
11 Print Average
```

There are four errors in this code.

Locate these errors and suggest a correction to remove each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

..... [8]

71 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]

72 **(a)** Write an algorithm to input three different numbers, and then output the largest number. Use **either** pseudocode **or** a flowchart.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[4]

(b) Give **two** sets of test data to use with your algorithm in **part (a)** and explain why you chose each set.

Test data set 1 .....

Reason .....

.....

.....

Test data set 2 .....

Reason .....

.....

.....

[4]

73 An algorithm has been written in pseudocode to input 100 numbers and print out the sum. A `REPEAT ... UNTIL` loop has been used.

```
Count ← 0
Sum ← 0
REPEAT
  INPUT Number
  Sum ← Sum + Number
  Count ← Count + 1
UNTIL Count > 100
PRINT Sum
```

(a) Find the error in the pseudocode and suggest a correction.

Error.....

Correction .....

.....

[2]

(b) Rewrite the correct algorithm using a more suitable loop structure.

.....

.....

.....

.....

.....

.....

.....[3]

74 This section of program code asks for 80 numbers between 100 and 1000 to be entered. It checks that the numbers are in the correct range, and stores them in an array. It counts how many of the numbers are larger than 500 and then outputs the result when the program is finished.

```
1 Count = 0
2 FOR Index = 1 TO 80
3   INPUT 'Enter a number between 100 and 1000', Number
4   WHILE Number = 99 AND Number = 1001
5     INPUT 'This is incorrect, please try again', Number
6   ENDWHILE
7   Num[80] = Number
8   IF Number > 500 THEN Count = Count + 1
9  UNTIL Index = 80
10 PRINT Index
11 PRINT ' numbers were larger than 500'
```

There are **four** lines of code that contain errors.

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]

75 Write an algorithm using **either** pseudocode **or** a flowchart, to:

- input a positive integer
- use this value to set up how many other numbers are to be input
- input these numbers
- calculate and output the total and the average of these numbers.

76 **(a)** Write an algorithm to input 1000 numbers. Count how many numbers are positive and how many numbers are zero. Then output the results. Use **either** pseudocode **or** a flowchart.

(b) Give one change you could make to your algorithm to ensure initial testing is more manageable.

..... [1]

77 Explain the difference between the programming concepts of **counting** and **totalling**. Include an example of a programming statement for each concept in your explanation.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

78 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]

79 This section of program code reads the contents of the array, totals the numbers and prints out the sum and average of the numbers. Assume the array is full.

Complete the **four** missing items by writing them in the spaces provided in this code.

```
1   Numbers[1:30]
2   Total = 0
3   ..... = 0
4   FOR Count = 1 TO .....
5   Number = Numbers[Count]
6   Total = ..... + Number
7   Counter = Counter + 1
8   ..... Count
9   PRINT 'The sum of the numbers you entered is ', Number
10  PRINT 'The average of the numbers you entered is ', Number / Counter
```

[4]

80 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]

(b) **(i)** The algorithm only allows one attempt at inputting an acceptable value.

State how you would change the algorithm so that it continues until a suitable input is supplied.

.....

.....[1]

(ii) Re-write the algorithm in full, using pseudocode, to implement your answer to **part (b)(i).**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

81 An algorithm is written in pseudocode:

```
Total ← 0
FOR Count ← 1 TO 50
  INPUT Num
  Total ← Total + Num
NEXT Count
OUTPUT Total
```

(a) Describe the purpose of the algorithm.

.....

.....

.....

.....

.....

.....[3]

(b) Re-write the algorithm in pseudocode using a different type of loop.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

(c) Describe how you could modify the original algorithm shown at the start of question 4, to allow any number of inputs.

.....

.....

.....

.....[2]

82 **(a)** Write an algorithm, using pseudocode, to input three different numbers, multiply the two larger numbers together and output the result. Use the variables: `Number1`, `Number2` and `Number3` for your numbers and `Answer` for your result.

[5]

(b) Give **two** sets of test data to use with your algorithm in **part (a)** and explain why you chose each set.

Test data set 1 .....

Reason .....

.....

Test data set 2 .....

Reason .....

.....

[4]

0478/22/O/N/18

83 This is a section of program code.

```
1 Total = 100.00
2 PRINT 'Enter the height of each member of your class, one at a
  time, when prompted'
3 FOR Count = 1 TO 30
4   PRINT 'Enter a height in metres'
5   INPUT Height
6   Total = Total + Height
7   PRINT Total / 30
8   Count = Count + 1
9 NEXT Count
```

(a) There are **three** errors in this code.

State the line numbers that contain the errors and describe how to correct each error.

Error 1 .....

.....

.....

Error 2 .....

.....

.....

Error 3 .....

.....

.....

[3]

(b) State the purpose of this program.

.....

.....

.....[1]

84 **(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]

(b) Show how you would change the corrected algorithm to total the numbers and print the total. Use a variable `Total`.

.....

.....

.....

.....

.....

.....

.....

..... [4]

85 **(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
```

Error 1 .....

.....

Correction .....

.....

Error 2 .....

.....

Correction .....

.....

Error 3 .....

.....

Correction .....

.....

Error 4 .....

.....

Correction .....

.....

[4]

(b) Describe how you would change the corrected algorithm to calculate the number accepted instead of rejected, using a variable `Accept`, and print a warning if fewer than 50% are accepted.

.....

.....

.....

.....

.....

.....

.....

..... [4]

86 Examine the following pseudocode:

```
INPUT A
INPUT B
INPUT C
INPUT D
INPUT E
INPUT F
INPUT G
INPUT H
INPUT I
INPUT J
INPUT K
INPUT L
T ← A + B + C + D + E + F + G + H + I + J + K + L
OUTPUT "The average equals ", T/12
```

(a) Describe what happens in this pseudocode.

.....

.....

.....

.....

.....

.....[3]

(b) Describe how this pseudocode could be altered to allow any number of values to be input.

.....

.....

.....

.....

.....

.....

.....

.....[3]

(c) Re-write the given pseudocode to allow any number of values to be input.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[5]

87 An algorithm has been written in pseudocode to select a random number using the function `RandInt(n)`, which returns a whole number between 1 and the argument n. The algorithm then allows the user to guess the number.

```
Number ← RandInt(100)
TotalTry ← 1
REPEAT
  PRINT "Enter your guess now, it must be a whole number"
  INPUT Guess
  IF TotalTry > Number
    THEN
      PRINT "Too large try again"
  ENDIF
  IF Guess > Number
    THEN
      PRINT "Too small try again"
  ENDIF
  TotalTry ← Guess + 1
UNTIL Guess <> Number
TotalTry ← TotalTry - 1
PRINT "Number of guesses ", TotalTry
```

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

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

88 A programmer writes a program to weigh baskets of fruit in grams, keeping a total of the weight and counting the number of baskets. The total weight is stored in a variable `Total` and the number of baskets is stored in a variable `BasketCount`.

Explain, including examples of programming statements, how totalling and counting could be used in this program.

Totalling .....

.....

.....

.....

.....

.....

Counting .....

.....

.....

.....

.....

.....

[4]

89 The following pseudocode algorithm uses nested IF statements.

```
IF Response = 1
  THEN
    X ← X + Y
  ELSE
    IF Response = 2
      THEN
        X ← X - Y
      ELSE
        IF Response = 3
          THEN
            X ← X * Y
          ELSE
            IF Response = 4
              THEN
                X ← X / Y
              ELSE
                OUTPUT "No response"
            ENDIF
        ENDIF
    ENDIF
ENDIF
```

(a) Name the type of statement demonstrated by the use of `IF … THEN … ELSE … ENDIF`

.....

..... [1]

(b) Re-write the pseudocode algorithm using a `CASE` statement.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

90 The pseudocode algorithm shown should allow numbers to be entered and should allow 50 numbers to be stored in an array.

```
Count ← 0
REPEAT
  INPUT Values[Count]
  Count ← Count + 1
UNTIL Count = 0
```

(a) Explain why the algorithm will never end.

.....

.....

.....

.....

..... [2]

(b) Re-write the original pseudocode so that it terminates correctly **and** also prevents numbers below 100 from being stored in the array `Values[ ]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Describe how you could change your pseudocode in **part (b)** so that it prevents numbers below 100 and above 200 from being stored in the array `Values[ ]`

.....

.....

.....

..... [2]

91 **(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.

```
01  HighestMark ← 100
02  HighestMarkStudents ← 0
03  FOR Count ← 1 TO 35
04      OUTPUT "Please enter student name"
05      INPUT Name[Count]
06      OUTPUT "Please enter student mark"
07      INPUT Mark[Counter]
08      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]

(b) Explain how you could extend the algorithm to also find the lowest mark awarded, count the number of students with that mark, and output both these values.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... [6]

92 Draw a line to connect each **Description** to the most appropriate **Pseudocode example**.

Description	Pseudocode example
A loop that will iterate at least once	`CASE … OF … OTHERWISE … ENDCASE`
A loop that will not be executed on the first test if the condition is false	`Number ← Number + 1`
A conditional statement	`WHILE … DO … ENDWHILE`
Totalling	`Sum ← Sum + NewValue`
Counting	`REPEAT … UNTIL`

[4]

93 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]

94 Describe the purpose of variables and constants. Use an example of each in your answer.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

95 This pseudocode algorithm calculates the weight and number of bags in a load of firewood. The weight in kilograms of each bag is input. The algorithm finishes when either 50 bags have been weighed, or as soon as the total weight exceeds 1000 kilograms. Only then are the total weight and the number of bags in the load output.

```
01  TotalWeight ← 1000
02  BagCount ← 0
03  MaxBag ← 50
04  MaxWeight ← 1000
05  REPEAT
06      OUTPUT "Please Enter weight of bag"
07      INPUT Weight
08      TotalWeight ← TotalWeight + Weight
09      BagCount ← BagCount + 1
10      OUTPUT "Number of bags in the load is ", BagCount
11  UNTIL TotalWeight > MaxWeight AND BagCount >= MaxBag
12  OUTPUT "Total weight of the load is ", MaxWeight
```

(a) Give the line number(s) from the algorithm of:

an assignment statement .....

a loop .....

a counting statement .....

a totalling statement .....

[4]

(b) Give the line numbers of the **four** errors in this 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]

(c) Explain how you could extend the algorithm to calculate and display the average weight of a bag of firewood in the load.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

96 This pseudocode algorithm is used as a validation check.

```
PRINT "Input a number from 1 to 5000"
REPEAT
  INPUT Number
  IF Number < 1 OR Number > 5000
    THEN
      PRINT "Invalid number, please try again"
  ENDIF
UNTIL Number >= 1 AND Number <= 5000
PRINT Number, " is within the correct range"
```

Identify **three** different types of test data. For each type, give an example of the test data you would use to test this algorithm and state a reason for your choice of test.

Type of test data 1 .....

Test data .....

Reason .....

.....

Type of test data 2 .....

Test data .....

Reason .....

.....

Type of test data 3 .....

Test data .....

Reason .....

.....

[6]

97 An algorithm has been written in pseudocode to check the temperature readings taken from a freezer are within the range –18 degrees to –25 degrees inclusive.

The algorithm counts the number of times that the temperature reading is below –25 degrees and the number of times that the temperature reading is above –18 degrees.

An engineer is called if there are more than 10 temperature readings below –25 degrees.

An alarm sounds if there are more than 5 temperature readings above –18 degrees.

```
01  TooHot ← 0
02  TooCold ← 1000
03  REPEAT
04      OUTPUT "Please enter temperature"
05      INPUT Temperature
06      IF Temperature < -25
07        THEN
08          TooCold ← TooCold - 1
09      ENDIF
10      IF Temperature > -18
11        THEN
12          TooHot ← TooHot + 1
13      ENDIF
14  UNTIL TooHot > 5 OR TooCold > 10
15  IF TooHot < 5
16    THEN
17      INPUT "Alarm!!"
18  ENDIF
19  IF TooCold > 10
20    THEN
21      OUTPUT "Call the Engineer"
22  ENDIF
```

(a) Give the line number(s) from the algorithm of:

an assignment statement .....

a loop .....

a counting statement .....

a selection statement .....

[4]

(b) 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]

(c) Explain how you could extend the algorithm to count the number of times the temperature readings are within the range –18 degrees to –25 degrees inclusive.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

98 The pseudocode algorithm shown has been written by a teacher to enter marks for the students in her class and then to apply some simple processing.

```
Count ← 0
REPEAT
  INPUT Score[Count]
  IF Score[Count] >= 70
    THEN
      Grade[Count] ← "A"
    ELSE
      IF Score[Count] >= 60
        THEN
          Grade[Count] ← "B"
        ELSE
          IF Score[Count] >= 50
            THEN
              Grade[Count] ← "C"
            ELSE
              IF Score[Count] >= 40
                THEN
                  Grade[Count] ← "D"
                ELSE
                  IF Score[Count] >= 30
                    THEN
                      Grade[Count] ← "E"
                    ELSE
                      Grade[Count] ← "F"
                  ENDIF
              ENDIF
          ENDIF
      ENDIF
  ENDIF
  Count ← Count + 1
UNTIL Count = 30
```

(a) Describe what happens in this algorithm.

.....

.....

.....

.....

.....

.....

.....

..... [3]

(b) Write the pseudocode to output the contents of the arrays `Score[]` and `Grade[]` along with suitable messages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

(c) Describe how you could change the algorithm to allow teachers to use it with any size of class.

.....

.....

.....

.....

.....

.....

..... [3]

99 **(a)** Write an algorithm in pseudocode to input 500 positive whole numbers. Each number input must be less than 1000. Find and output the largest number input, the smallest number input and the range (difference between the largest number and smallest number).

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [6]

(b) Describe how the algorithm could be changed to make testing less time-consuming.

.....

.....

.....

..... [2]

100 The pseudocode algorithm should allow a user to input the number of scores to be entered and then enter the scores. The scores are totalled, the total is output and the option to enter another set of scores is offered.

```
1   Count ← 0
2   REPEAT
3     FullScore ← 20
4     INPUT Number
5     FOR StoreLoop ← 1 TO Number
6       INPUT Score
7       FullScore ← FullScore
8     UNTIL StoreLoop = Number
9     OUTPUT "The full score is ", FullScore
10    OUTPUT "Another set of scores (Y or N)?"
11    OUTPUT Another
12    IF Another = "N"
13      THEN
14        Count ← 1
15    ENDIF
16  UNTIL Count = 1
```

(a) Identify 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]

(b) Show how you could change the algorithm to store the individual scores in the array `ScoreArray[]`, then find and print the average score once the scores have all been entered.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

101 An algorithm has been written in pseudocode to generate 50 positive random integers with values less than or equal to 100. These random integers are stored in the array `RandNum[ ]`

The function `Rand(X, Y)` generates a random integer greater than or equal to `X` and less than `Y`. For example, `Rand(1, 4)` generates `1` or `2` or `3`.

```
1 Count ← 0
2 REPEAT
3     RandNum[Counter] ← Rand(1, 100)
4     Count ← Count + 2
5 UNTIL Count <= 50
```

(a) Find the **four** errors in the pseudocode and write a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

.....

[4]

(b) The pseudocode for this algorithm could be shortened by the use of a `FOR ... NEXT` loop.

Rewrite the algorithm using a `FOR ... NEXT` loop.

.....

.....

.....

..... [3]

(c) Identify another loop structure available in pseudocode.

..... [1]

102 The pseudocode algorithm should work as a calculator and output the result.

```
1   Continue ← 1
2   WHILE Continue = 0
3     OUTPUT "Enter 1 for +, 2 for -, 3 for * or 4 for /"
4     INPUT Operator
5     OUTPUT "Enter the first value"
6     INPUT Value1
7     OUTPUT "Enter the second value"
8     OUTPUT Value2
9     IF Operator
10      1: Answer ← Value1 + Value2
11      2: Answer ← Value1 - Value2
12      3: Answer ← Value1 * Value2
13      4: Answer ← Value1 / Value2
14    ENDCASE
15    OUTPUT "The answer is ", Value1
16    OUTPUT "Do you wish to enter more values (Yes or No)?"
17    INPUT MoreValues
18    IF MoreValues = "No"
19      THEN
20        Continue ← 1
21    ENDIF
22  UNTIL Continue = 0
```

(a) Find the **five** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

Error 5 .....

Correction .....

.....

(b) The algorithm needs changing to allow only the numbers 1, 2, 3, or 4 to be entered for the input variable `Operator`.

Write the pseudocode to perform this task and state where in the algorithm it would be located.

Pseudocode .....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Location in algorithm .....

.....

.....

[5]

103 An algorithm allows a user to input their password and checks that there are at least eight characters in the password. Then, the user is asked to re-input the password to check that both inputs are the same. The user is allowed three attempts at inputting a password of the correct length and a matching pair of passwords. The pre-defined function `LEN(X)` returns the number of characters in the string, `X`

```
01  Attempt ← 0
02  REPEAT
03    PassCheck ← TRUE
04    OUTPUT "Please enter your password "
05    INPUT Password
06    IF LEN(Password) < 8
07      THEN
08        PassCheck ← TRUE
09      ELSE
10        OUTPUT "Please re-enter your password "
11        INPUT Password2
12        IF Password <> Password
13           THEN
14           PassCheck ← FALSE
15        ENDIF
16    ENDIF
17    Attempt ← Attempt + 1
18  UNTIL PassCheck OR Attempt <> 3
19  IF PassCheck
20    THEN
21     OUTPUT "Password success"
22    ELSE
23      OUTPUT "Password fail"
24  ENDIF
```

(a) Identify the **three** errors in the pseudocode and suggest a correction to remove each error.

Error 1 .....

Correction .....

Error 2 .....

Correction .....

Error 3 .....

Correction .....

[3]

(b) The algorithm includes **two** types of check on the data input.
Identify and describe each type of check.

Type of check 1 .....

Description .....

.....

Type of check 2 .....

Description .....

.....

[4]

(c) Give **two** sets of test data for this algorithm and a reason for choosing each set.

Each set of test data and its reason must be different.

Set 1 .....

Reason .....

.....

Set 2 .....

Reason .....

.....

[4]

104 Describe what is meant by the terms variable and constant and give an example of each in your answer.

Variable .....

.....

.....

.....

Constant .....

.....

.....

.....

[4]

105 An algorithm has been written to:

- set 100 elements of the array `Reading[1:100]` to zero
- input integer values between 1 and 100
- end the process with an input of –1
- reject all other values
- count and output the number of times each value is input, starting with the largest value.

(a) Complete the pseudocode algorithm:

```
01  FOR Count ← 1 TO .....
02    Reading[Count] ← 0
03  NEXT Count
04  OUTPUT "Please enter next reading "
05  INPUT Value
06  WHILE Value <> -1 DO
07    IF Value <= 0 OR .....
08      THEN
09        OUTPUT "Reading out of range"
10      ELSE
11        Reading[Value] ← .....
12    ENDIF
13    OUTPUT "Please enter next reading "
14    .....
15  ENDWHILE
16  Count ← 100
17  REPEAT
18    OUTPUT "There are ", .....,
             " readings of ", Count
19    Count ← .....
20  UNTIL Count = 0
```

[6]

(b) Describe how the algorithm could be changed so that it does **not** output any counts of zero.

.....

.....

.....

.....

.....

..... [3]

106 This pseudocode should allow 500 marks to be entered into the algorithm. If the mark is 80 or greater it is stored in an array for higher marks. If the mark is less than 80, but greater than or equal to 50 it is stored in an array for middle marks. The remaining marks are stored in an array for lower marks. The results from the algorithm are displayed at the end.

```
01 HighList ← 0
02 MidList ← 0
03 LowList ← 0
04 MarksEntry ← 0
05 REPEAT
06   INPUT Mark
07   IF Mark >= 80
08     THEN
09       Higher[HighList] ← MarksEntry
10       HighList ← HighList + 1
11     ELSE
12       IF Mark >= 50
13         THEN
14           Middle[MidList] ← Mark
15           MidList ← MidList
16         ELSE
17           Lower[HighList] ← Mark
18           LowList ← LowList + 1
19       ENDIF
20   ENDIF
21   MarksEntry ← MarksEntry + 1
22 NEXT MarksEntry = 500
23 OUTPUT "You entered ", HighList, " higher marks"
24 OUTPUT "You entered ", MidList, " middle marks"
25 OUTPUT "You entered ", LowList, " lower marks"
```

(a) Identify 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]

(b) The corrected algorithm needs to be changed so that any number of marks may be entered and the algorithm runs until the user tells it to stop.

Write the new pseudocode statements that would be needed to achieve this and state where in the algorithm they would be placed.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

107 An algorithm has been written to:

- set all 50 elements of the array `Reading[1:50]` to zero
- input values between 35 and 50 inclusive
- end the process when an input of –1 is made or 50 valid numbers have been entered
- reject all other values
- count the number of times each valid value is input
- output the number of times each value has been input, starting with the lowest value.

(a) Complete the pseudocode algorithm:

```
01  FOR Count ← 1 TO ..........
02    Reading[Count] ← 0
03  NEXT Count
04  Count ← 1
05  OUTPUT "Please enter next reading "
06  INPUT Value
07  REPEAT
08    IF Value < 35 OR ..........
09      THEN
10        OUTPUT "Reading out of range"
11      ELSE
12        Reading[Value] ← ..........
13    Count = Count + 1
14    ENDIF
15    IF Count <= 50
16      THEN
17        OUTPUT "Please enter next reading "
18       ..........
19    ENDIF
20  UNTIL Value = -1 OR Count > 50
21  Count ← 35
22  REPEAT
23    OUTPUT "There are ", ..........,
            " readings of ", Count
24    Count ← ..........
25  UNTIL Count > 50
```

[6]

(b) Describe how the algorithm could be changed to output the number of times each value has been input, starting with the highest value.

.....

.....

.....

.....

.....

..... [3]

108 An algorithm has been written in pseudocode to allow some numbers to be input. All the positive numbers that are input are totalled and this total is output at the end.
An input of 0 stops the algorithm.

```
01 Exit ← 1
02 WHILE Exit <> 0 DO
03     INPUT Number
04     IF Number < 0
05       THEN
06         Total ← Total + Number
07       ELSE
08         IF Number = 0
09           THEN
10             Exit ← 1
11         ENDIF
12     ENDIF
13 ENDIF
14 OUTPUT "The total value of your numbers is ", Number
```

(a) Identify 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]

(b) Describe how you could change the corrected algorithm to record and output how many positive numbers have been included in the final total.

You do **not** need to rewrite the algorithm.

.....

.....

.....

.....

.....

.....

.....

..... [4]

109 An algorithm has been written in pseudocode to calculate a check digit for a four-digit number. The algorithm then outputs the five-digit number including the check digit.
The algorithm stops when –1 is input as the fourth digit.

```
01 Flag ← FALSE
02 REPEAT
03     Total ← 0
04     FOR Counter ← 1 TO 4
05       OUTPUT "Enter a digit ", Counter
06       INPUT Number[Counter]
07       Total ← Total + Number * Counter
08       IF Number[Counter] = 0
09         THEN
10           Flag ← TRUE
11       ENDIF
12     NEXT Counter
13     IF NOT Flag
14       THEN
15         Number[5] ← MOD(Total, 10)
16         FOR Counter ← 0 TO 5
17             OUTPUT Number[Counter]
18         NEXT
19     ENDIF
20 UNTIL Flag
```

(a) Give the line number(s) for the statements showing:

Totalling .....

Count-controlled loop .....

Post-condition loop .....

[3]

(b) Identify the **three** errors in the pseudocode and suggest a correction for each error.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(c) The algorithm does **not** check that each input is a single digit.
Identify the place in the algorithm where this check should occur.
Write pseudocode for this check.
Your pseudocode must make sure that the input is a single digit and checks for `-1`

Place in algorithm .....

Pseudocode .....

.....

.....

.....

.....

.....

[4]

110 An algorithm has been written in pseudocode to allow 100 positive numbers to be input. The total and the average of the numbers are output.

```
01 Counter ← 100
02 Total ← 0
03 WHILE Counter > 100 DO
04    INPUT Number
05    IF Number > 0
06       THEN
07         Total ← Total + Counter
08         Counter ← Counter + 1
09    ENDCASE
10 ENDWHILE
11 OUTPUT "The total value of your numbers is ", Total
12 OUTPUT "The average value of your numbers is ", Total / 100
```

(a) Identify the **four** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

(b) Describe the changes you should make to the corrected algorithm so that a count-controlled loop is used to allow 100 positive numbers to be input.

You do **not** need to rewrite the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

111 An algorithm has been written in pseudocode.

```
01 DECLARE A[1:10] : STRING
02 DECLARE T : STRING
03 DECLARE C, L : INTEGER
04 L ← 10
05 FOR C ← 1 TO L
06     OUTPUT "Please enter name "
07     INPUT A[C]
08 NEXT C
09 FOR C ← 1 TO L
10     FOR L ← 1 TO 9
11         IF A[L] > A[L + 1]
12           THEN
13             T ← A[L]
14             A[L] ← A[L + 1]
15             A[L + 1] ← T
16         ENDIF
17     NEXT L
18 NEXT C
19 FOR C ← 1 TO L
20     OUTPUT "Name ", C, " is ", A[C]
21 NEXT C
```

(a) State the purpose of this pseudocode algorithm.

.....

..... [1]

(b) State **four** processes in this algorithm.

1 .....

.....

2 .....

.....

3 .....

.....

4 .....

.....

[4]

(c) Meaningful identifiers have **not** been used in this algorithm.
Suggest suitable meaningful identifiers for:

The array:

A .....

The variables:

T .....

C .....

L .....

[3]

(d) State **two** other ways the algorithm can be made easier to understand and maintain.

1 .....

.....

2 .....

.....

[2]

112 An algorithm has been written in pseudocode to allow the names of 50 cities and their countries to be entered and stored in a two-dimensional (2D) array. The contents of the array are then output.

```
01 DECLARE City ARRAY[1:50, 1:2] OF BOOLEAN
02 DECLARE Count : INTEGER
03 DECLARE Out : INTEGER
04 Count ← 1
05 IF
06     OUTPUT "Enter the name of the city"
07     INPUT City[Count, 2]
08     OUTPUT "Enter the name of the country"
09     INPUT City[Count, 2]
10     Count ← Count + 1
11 UNTIL Count = 50
12 FOR Out ← 1 TO 1
13     OUTPUT "The city ", City[Out, 1], " is in ", City[Out, 2]
```

(a) Identify the **four** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

(b) Describe the changes you should make to the corrected algorithm to allow the name of a country to be input and to display only the stored cities from that country.

You do **not** need to rewrite the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

113 Explain how variables and constants should be used when creating and running a program.

.....

.....

.....

.....

.....

..... [3]

114 An algorithm has been written in pseudocode to check if a new password is in a list of previously used passwords `OldList[]`
If the password is **not** found, the new password will be stored at the end of the list to replace `"XXXX"` already stored there.

```
01 OUTPUT "Enter your new password "
02 INPUT NewPassword
03 Posn ← 1
04 Found ← FALSE
05 REPEAT
06     IF Password = OldList[Posn]
07       THEN
08         Found ← TRUE
09       ELSE Posn ← Posn + 1
10     ENDIF
11 UNTIL Found AND OldList[Posn] = "XXXX"
12 IF Found
13   THEN
14     OUTPUT "Password has been used before"
15   ELSE
16     INPUT "New password accepted"
17     OldList[Posn] ← NewPassword
18 ENDIF
```

(a) Identify the **three** errors in the pseudocode and suggest corrections.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

[3]

(b) Complete this flowchart for the corrected algorithm:

115 This pseudocode algorithm is intended to allow, at random, between 1 and 20 values to be entered and totalled. The total and average of the entered values are output at the end of the algorithm.

```
01 DECLARE Loop : STRING
02 DECLARE Limit : INTEGER
03 DECLARE Value : REAL
04 DECLARE Total : REAL
05 Total ← 0
06 Limit ← ROUND(RANDOM() * 19,0) + 1
07 IF Loop ← 1 TO Limit
08     OUTPUT "Enter a number"
09     INPUT Loop
10     Total ← Total * Value
11 NEXT Loop
12 OUTPUT "The total of the numbers entered is ", Total
13 OUTPUT "The average of the numbers entered is ", Total / Limit
```

(a) Identify the line numbers of **four** errors in the pseudocode and suggest corrections.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

Error 4 line number .....

Correction .....

.....

[4]

(b) Write the pseudocode statement that would output the average calculated in line 13 of the algorithm rounded to **one** decimal place.

.....

.....

.....

..... [2]

(c) Explain how you should alter the original corrected algorithm to make sure that all the numbers entered are between 1 and 500 inclusive. If any numbers fall outside these limits, a replacement value is requested and entered.

You do **not** need to re-write the algorithm.

.....

.....

.....

.....

.....

.....

.....

..... [4]

116 Write the pseudocode statements to perform this task:

- accept the input of a whole number from 1 to 4 inclusive
- use a `CASE` statement to:
  - output the number (1 to 4 inclusive) that was entered
  - output the word "ERROR" if a 1 to 4 inclusive number was **not** entered.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

117 **(a)** Outline why it is useful to store data in a file.

.....

.....

.....

..... [2]

(b) The function `LENGTH(X)` calculates the length of a string `X`

Write the pseudocode statements to:

- read the contents of the text file `Quotation.txt` into an appropriate string variable that has been declared
- output the string in upper case and the length of the string.

.....

.....

.....

.....

.....

.....

.....

..... [4]

118 A program needs to make sure the characters input for a product code meet these rules:

- The product code is six characters in length.
- The first two characters must be "PD".
- The last four characters must be a number in the range 1000 to 9999 inclusive.

(a) Identify **three** validation checks and state how each check would make sure the product code met one of these rules.

Check 1 .....

.....

.....

Check 2 .....

.....

.....

Check 3 .....

.....

.....

[6]

(b) The program design will include a pseudocode algorithm. Assume that the product code is stored in the variable `Product`

(i) Write the pseudocode to make sure that the product code is six characters in length.

.....

.....

.....

.....

..... [2]

(ii) Write the pseudocode to make sure that the first two characters of the product code are "PD".

.....

.....

.....

.....

..... [2]

119 Totalling and counting are standard methods of solution.

Numbers are input. The number 9999.9 is the last number to be input and is ignored.

(a) Write an algorithm in pseudocode to total the numbers input and to output the total. You do **not** need to validate the input.

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Write an algorithm in pseudocode to count and output the number of input values that are greater than 100. You do **not** need to validate the input.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

120 An algorithm has been written in pseudocode to find and display the maximum and minimum values in an array of 1000 positive numbers. The array `List[]` starts at index `1`

```
01 Max ← List[1]
02 Min ← List[1]
03 FOR Counter ← 2 TO 1000
04     IF List[Counter] < Max
05       THEN
06         Max ← List[Counter]
07     ENDIF
08     IF List[Count] < Min
09       THEN
10         Min ← List[Counter]
11     ENDWHILE
12 NEXT Counter
13 OUTPUT "Maximum value is ", Max
14 OUTPUT "Minimum value is ", Min
```

(a) Give a line number for each of these types of statement:

Assignment statement .....

Selection statement .....

Iteration statement .....

[3]

(b) Identify the line numbers of the **three** errors in the pseudocode and suggest a correction for each error.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

[3]

121 This pseudocode algorithm is intended to allow data for up to 50 people to be entered and stored in a two-dimensional (2D) array. The data is their last name, first name and the city in which they live.

```
01 DECLARE People : ARRAY[1:50, 1:3] OF REAL
02 DECLARE Count : INTEGER
03 DECLARE Response : CHAR
04 DECLARE Continue : BOOLEAN
05 FOR I ← 1 TO 50
06     FOR J ← 1 TO 3
07         People[I, J] ← ""
08     NEXT J
09 NEXT I
10 Count ← 100
11 Continue ← TRUE
12 CASE OF
13     OUTPUT "Enter the last name"
14     INPUT People[Count, 1]
15     OUTPUT "Enter the first name"
16     INPUT People[Count, 2]
17     OUTPUT "Enter the city"
18     INPUT People[Count, 3]
19     OUTPUT "Do you want to enter another name (Y or N)?"
20     INPUT Response
21     IF Response = 'N'
22       THEN
23         Continue ← FALSE
24       ELSE
25         Count ← Count + 1
26     ENDIF
27 UNTIL NOT Count
```

(a) Identify the line numbers of the **four** errors in the pseudocode and suggest corrections.

Error 1 line number .....

Correction .....

.....

Error 2 line number .....

Correction .....

.....

Error 3 line number .....

Correction .....

.....

Error 4 line number .....

Correction .....

.....

[4]

(b) Write the pseudocode that you could add to the end of this algorithm to output the contents of the array. Make sure that the output ends when the data in the array ends.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(c) Explain how you could alter the original corrected algorithm to make sure that the number of elements being added to the array does not exceed the maximum size of the array (50 elements).

.....

.....

.....

.....

.....

.....

.....

..... [4]

122 An incomplete algorithm has been written in pseudocode to count the number of zeros stored in an array and total the non-zero values.

```
01 DECLARE A[1:50] : INTEGER
02 DECLARE C : INTEGER
03 DECLARE I : INTEGER
04 DECLARE T : INTEGER
05 I ← 0
06 .....
07 FOR C ← 1 TO 50
08     IF A[C] .....
09       THEN
10         T ← T + 1
11       ELSE
12         I ← I + A[C]
13     ENDIF
14 .....
```

(a) Complete the given pseudocode algorithm. [3]

(b) Write the pseudocode to display, with suitable messages, the number of zeros stored in the array and the total of the non-zero values.

.....

.....

.....

.....

.....

..... [3]

(c) Meaningful identifiers have **not** been used in the algorithm.
Suggest suitable meaningful identifiers for:

The array:

`A` .....

The variables:

`T` .....

`C` .....

`I` .....

[3]

123 A programmer is designing a program to check the length of a password and to check if the password input is the same as the stored password.

The program requirements are:

- input the password, `Password`
- check if there are at least 8 characters in the password
- check that the password is **not** the same as the stored password `OldPass`
- output 'accepted' if both tests are completed successfully
- otherwise, output 'rejected'.

Use the variable names given.

(a) Complete the flowchart for the program.

[6]

(b) The accepted password, `Password`, is to be written to the file `MyPassword.txt`

Write pseudocode to:

- open the file
- write the accepted password to the file
- close the file.

.....

.....

.....

.....

..... [3]

(c) Explain why the accepted password needs to be stored in a file.

.....

.....

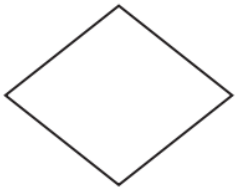
.....

..... [2]

124 **Four** flowchart symbols and **five** purposes are shown.

(a) Draw **one** line to link each flowchart symbol to its correct purpose.

Not all purposes will be used.

Flowchart symbol	Purpose
	subroutine
	process
	flow
	decision
	terminator

[4]

(b) An algorithm needs to total 50 numbers between 1 and 100 inclusive.

Draw a flowchart that:

- uses a count-controlled loop from 1 to 50
- uses an appropriate prompt to ask for a number between 1 and 100
- totals the numbers as they are entered
- outputs the total after the loop has completed with an appropriate message.

[6]

125 This pseudocode algorithm is intended to sort a pre-populated one-dimensional (1D) array named `ItemList` into alphabetical order using a bubble sort.

```
01 DECLARE ItemList : ARRAY[1:100] OF STRING
02 DECLARE Counter : STRING
03 DECLARE Limit : INTEGER
04 DECLARE Pass : INTEGER
05 DECLARE Swapped : BOOLEAN
06 DECLARE Temp : STRING
07 Limit ← 100
08 Pass ← 1
09 Temp ← TRUE
10 WHILE Swapped = TRUE OR Pass <= Limit - 1 DO
11     Swapped ← FALSE
12     FOR Counter ← 1 TO Limit - Pass
13         IF ItemList[Counter] > ItemList[Counter + 1]
14           THEN
15             Temp ← ItemList[Counter]
16             ItemList[Counter] ← ItemList[Counter + 1]
17             ItemList[Counter] ← Temp
18             Swapped ← TRUE
19         ENDCASE
20         Pass ← Pass + 1
21     NEXT Counter
22 ENDWHILE
```

(a) Identify the line numbers of **five** errors in the pseudocode and 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 .....

.....

Error 5 line number .....

Correction .....

.....

[5]

(b) A bubble sort algorithm can be written to include features that make it more efficient.

Explain why the **corrected** bubble sort algorithm is efficient.

.....

.....

.....

.....

.....

..... [3]

126 A programmer is testing a program that requires a positive value between 1 and 100 inclusive to be entered. The range check in the program is to be tested.

Identify **three** different types of test data to be used.

For each type of test data, give an example of the value(s) to be used and the expected outcome.

Type 1 .....

Example .....

Outcome .....

.....

Type 2 .....

Example .....

Outcome .....

.....

Type 3 .....

Example .....

Outcome .....

.....

[9]

TRACETABLE

127 **(a)** This pseudocode inputs an integer. The predefined function `DIV` gives the value of the division, e.g. `Y ← 10 DIV 3` gives the value `Y = 3`. The predefined function `MOD` gives the value of the remainder, e.g. `Y ← 10 MOD 3` gives the value `Y = 1`.

```
INPUT X
WHILE X > 15
  DO
  T1 ← X DIV 16
  T2 ← X MOD 16
  CASE T2 OF
    10:OUTPUT A
    11:OUTPUT B
    12:OUTPUT C
    13:OUTPUT D
    14:OUTPUT E
    15:OUTPUT F
    OTHERWISE OUTPUT T2
  ENDCASE
  X ← T1
ENDWHILE
CASE X OF
  10:OUTPUT A
  11:OUTPUT B
  12:OUTPUT C
  13:OUTPUT D
  14:OUTPUT E
  15:OUTPUT F
  OTHERWISE OUTPUT X
ENDCASE
```

Complete a trace table for each of the **two** input values 37 and 191.

Trace table for input value 37

X	T1	T2	OUTPUT

Trace table for input value 191

X	T1	T2	OUTPUT

[4]

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

.....

.....[2]

128 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

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

[5]

(b) Explain how you would change the algorithm to input eight digits (seven digits and the check digit) and output if the check digit entered is correct or not.

.....

.....

.....

.....

.....

.....

..... [3]

129 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

[3]

(b) Show how you could improve the algorithm written in pseudocode by writing an alternative type of conditional statement in pseudocode.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....[3]

130 The algorithm allows a number to be entered. It then calculates and outputs the next number in the mathematical series.

```
Fib ← 1
Prev2 ← 0
Prev1 ← 1
INPUT Number
IF Number = 0
    THEN Fib ← 0
ENDIF
WHILE Number > 2
    Fib ← Prev2 + Prev1
    Prev2 ← Prev1
    Prev1 ← Fib
    Number ← Number - 1
ENDWHILE
OUTPUT Fib
```

(a) Complete the trace table for the input data: **7**

Fib	Prev2	Prev1	Number	OUTPUT

[4]

(b) Complete the trace table for the input data: **2**

Fib	Prev2	Prev1	Number	OUTPUT

[2]

131 **(a)** Complete the trace table for this algorithm using the given input data.

```
Index ← 0
FOR Count ← 0 TO 7
  INPUT Value
  IF Value > 50
    THEN
      PassMarks[Index] ← Value
      Index ← Index + 1
  ENDIF
NEXT Count
PRINT "Number passed ", Index
```

Input data: 58, 40, 67, 85, 12, 13, 75, 82

Index	Count	Value	PassMarks								OUTPUT
			[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	

[6]

(b) Give the purpose of the algorithm shown in **part (a)**.

.....

.....

.....[1]

132 The algorithm performs an operation on the array named `MyData`

`DIV` means integer division, so only the whole number part of the result is returned
e.g. `7 DIV 2` returns a value of `3`

```
First ← 0
Last ← 16
Found ← FALSE
INPUT UserIn
WHILE (First <= Last) AND (Found = FALSE) DO
  Middle ← (First + Last) DIV 2
  IF MyData[Middle] = UserIn
    THEN
      Found ← TRUE
    ELSE
      IF UserIn < MyData[Middle]
        THEN
          Last ← Middle - 1
        ELSE
          First ← Middle + 1
      ENDIF
  ENDIF
ENDWHILE
OUTPUT Found
```

This table shows the contents of the array: `MyData` e.g. `MyData[2]` stores the value `5`

	MyData																
Index	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	[16]
Value	2	3	5	6	8	10	12	13	14	16	18	20	25	27	29	34	36

(a) Complete the trace table for the input data: 10

First	Last	UserIn	Middle	Found	OUTPUT

[6]

(b) Describe the function being performed by the algorithm.

.....

.....

.....

..... [2]

133 This algorithm finds prime numbers.

The pre-defined function `DIV` gives the value of the result of integer division, for example, `y ← 9 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

[5]

134 This pseudocode represents an algorithm.

```
REPEAT
  Flag ← 0
  FOR Count ← 0 to 3
    IF Num[Count] < Num[Count + 1]
      THEN
        Store ← Num[Count]
        Num[Count] ← Num[Count + 1]
        Num[Count + 1] ← Store
        Flag ← 1
    ENDIF
  NEXT Count
UNTIL Flag = 0
```

(a) The contents of the array at the start of the algorithm are:

Num[0]	Num[1]	Num[2]	Num[3]	Num[4]
45	56	30	12	15

Complete the trace table for the algorithm using the data given in the array.

Flag	Count	Num[0]	Num[1]	Num[2]	Num[3]	Num[4]	Store
		45	56	30	12	15	

[5]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

135 This algorithm checks passwords.

- Each password must be 8 or more characters in length; the predefined function `Length` returns the number of characters.
- Each password is entered twice, and the two entries must match.
- Either `Accept` or `Reject` is output.
- An input of 999 stops the process.

```
REPEAT
  OUTPUT "Please enter password"
  INPUT Password
  IF Length(Password) >= 8
    THEN
      INPUT PasswordRepeat
      IF Password <> PasswordRepeat
        THEN
          OUTPUT "Reject"
        ELSE
          OUTPUT "Accept"
      ENDIF
    ELSE
      OUTPUT "Reject"
  ENDIF
UNTIL Password = 999
```

(a) Complete the trace table for the algorithm using this input data:
Secret, Secret, VerySecret, VerySecret, Pa55word, Pa55word, 999, 888

Password	PasswordRepeat	OUTPUT

[3]

(b) Explain how the algorithm could be extended to allow three attempts at inputting the matching password. Any pseudocode statements used in your answer must be fully explained.

.....

.....

.....

.....

.....

.....

.....

..... [4]

136 The pseudocode represents an algorithm.

The pre-defined function `DIV` gives the value of the result of integer division.
For example, `Y = 9 DIV 4` gives the value `Y = 2`

The pre-defined function `MOD` gives the value of the remainder of integer division.
For example, `R = 9 MOD 4` gives the value `R = 1`

```
First ← 0
Last ← 0
INPUT Limit
FOR Counter ← 1 TO Limit
  INPUT Value
  IF Value >= 100
    THEN
      IF Value < 1000
        THEN
          First ← Value DIV 100
          Last ← Value MOD 10
            IF First = Last
              THEN
                OUTPUT Value
            ENDIF
      ENDIF
  ENDIF
NEXT Counter
```

(a) Complete the trace table for the algorithm using this input data:

8, 66, 606, 6226, 8448, 642, 747, 77, 121

Counter	Value	First	Last	Limit	OUTPUT

[5]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

137 The pseudocode represents an algorithm.

The pre-defined function `DIV` gives the value of the result of integer division.
For example, `Y = 11 DIV 4` gives the value `Y = 2`

```
Count ← 0
INPUT Limit
FOR In ← 1 TO Limit
  Logic ← TRUE
  Test ← 2
  INPUT Number
  REPEAT
    IF Number / Test = Number DIV Test
      THEN
        Logic ← FALSE
      ELSE
        Test ← Test + 1
    ENDIF
  UNTIL NOT Logic OR Test >= Number DIV 2
  IF Logic
    THEN
      Store[Count] ← Number
      Count ← Count + 1
  ENDIF
NEXT In
FOR Out ← 0 TO Count - 1
  OUTPUT Store[Out]
NEXT Out
```

(a) Complete the trace table for the algorithm using this input data:
5, 9, 5, 8, 10, 7

In	Logic	Test	Number	Store [Count]	Count	Limit	Out	OUTPUT

[7]

(b) State the purpose of this algorithm.

.....

.....

.....

..... [2]

(c) This algorithm only works for numbers that are 3 or greater.

Describe how you could change this algorithm to make sure that only numbers that are 3 or greater are entered. Any pseudocode statements used in your answer must be fully described.

.....

.....

.....

.....

.....

.....

.....

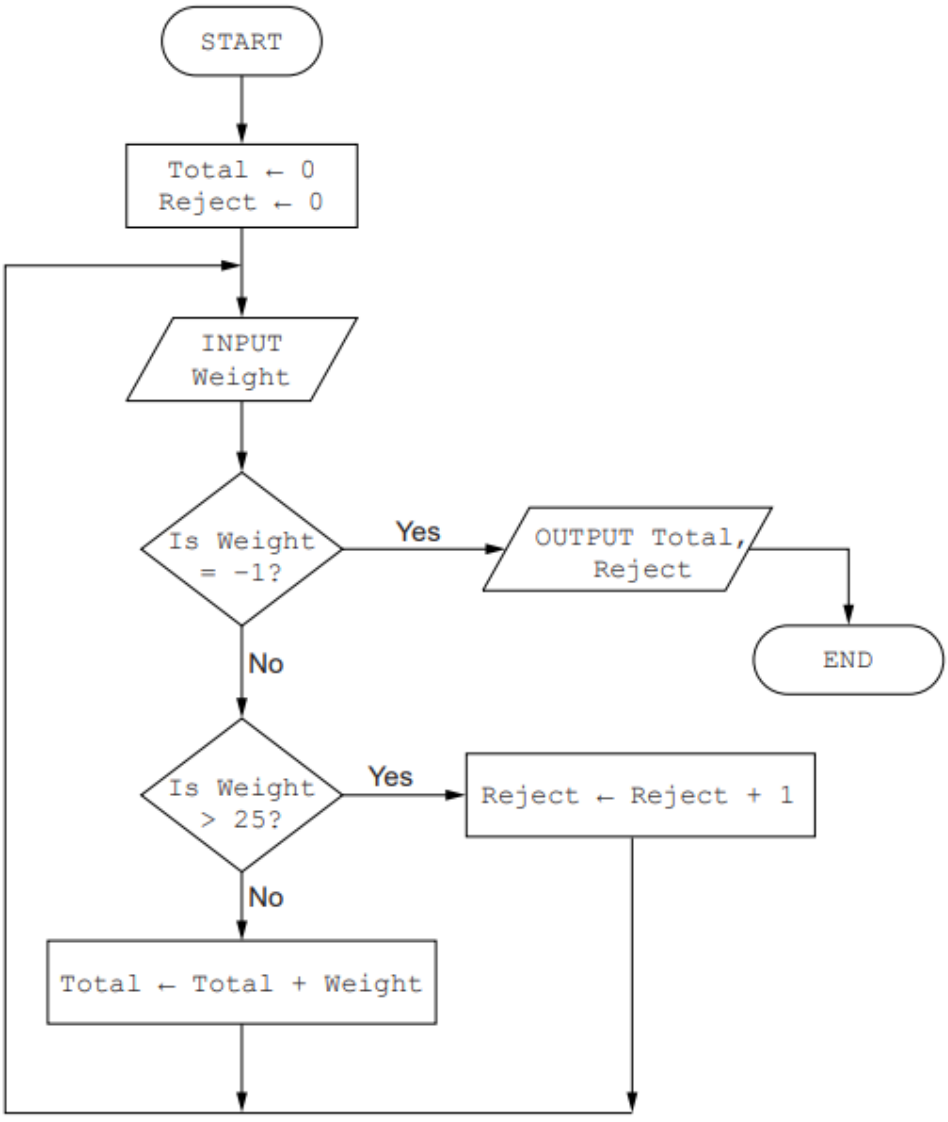
.....

.....

..... [3]

138 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.



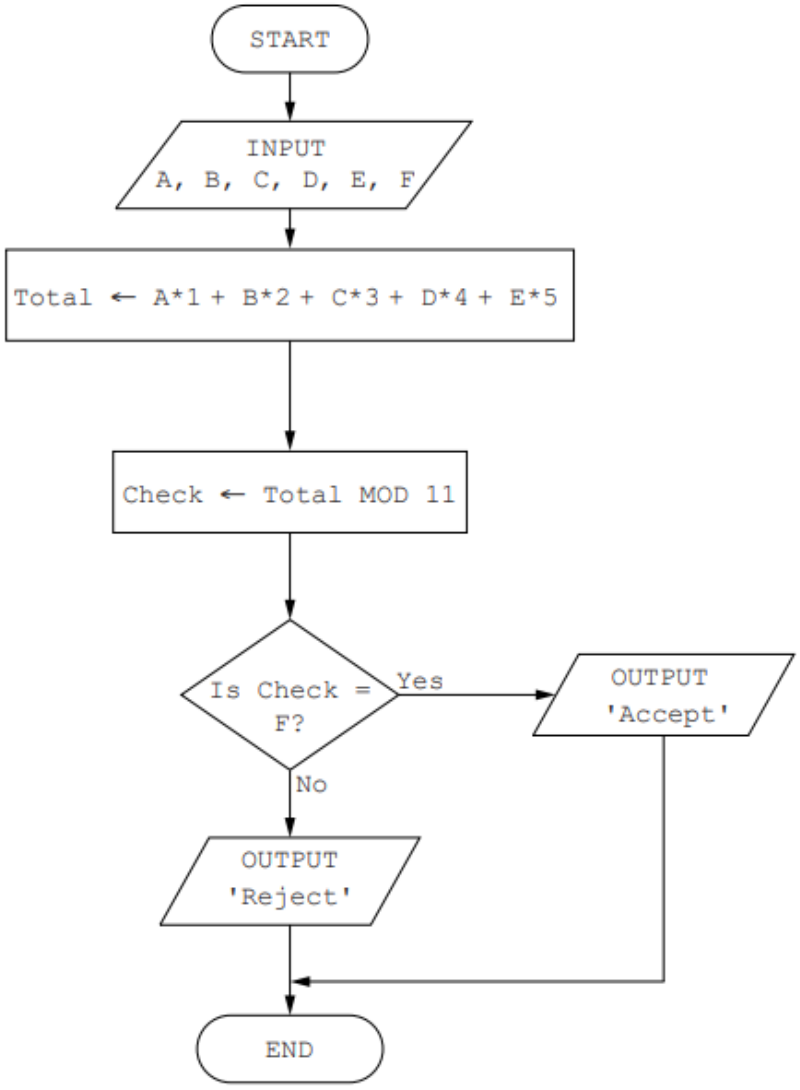
Complete the trace table for the input data:

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]

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



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 .....

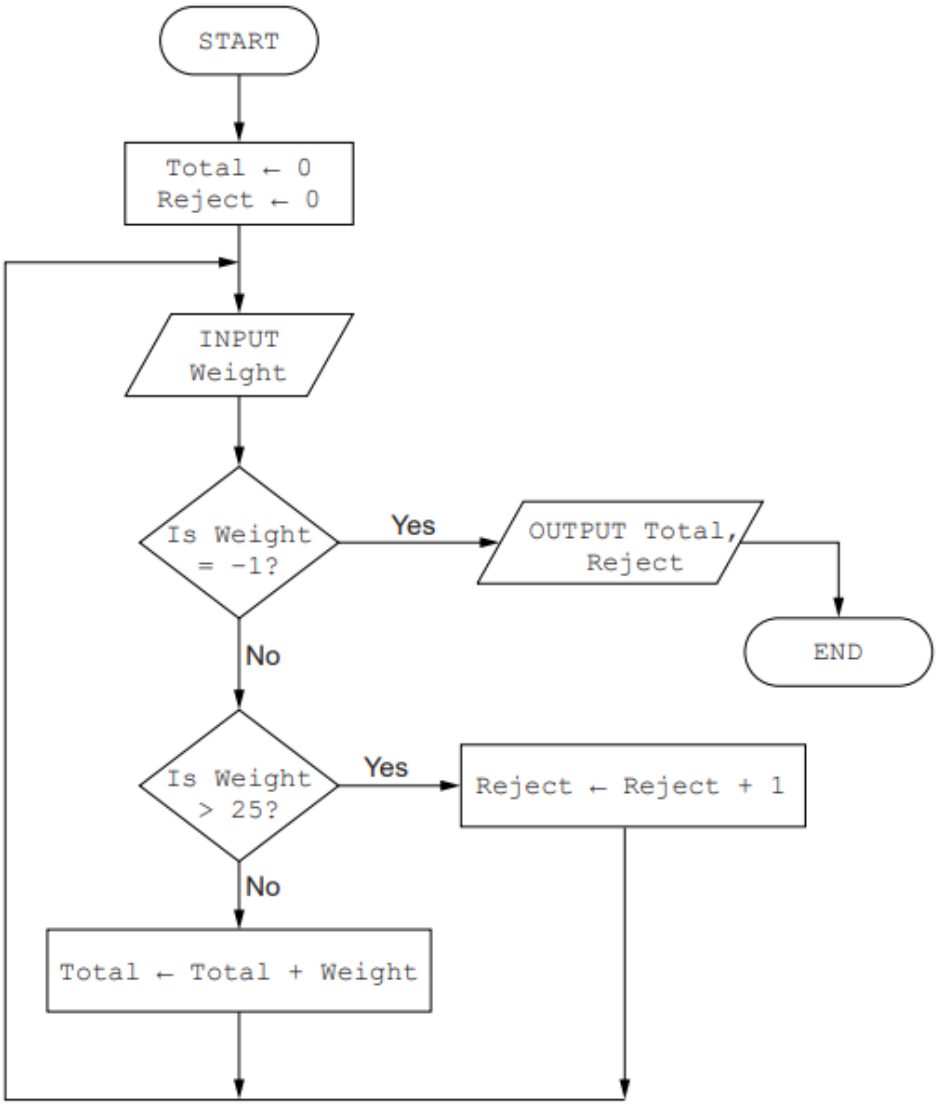
.....

.....

.....[3]

140 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.



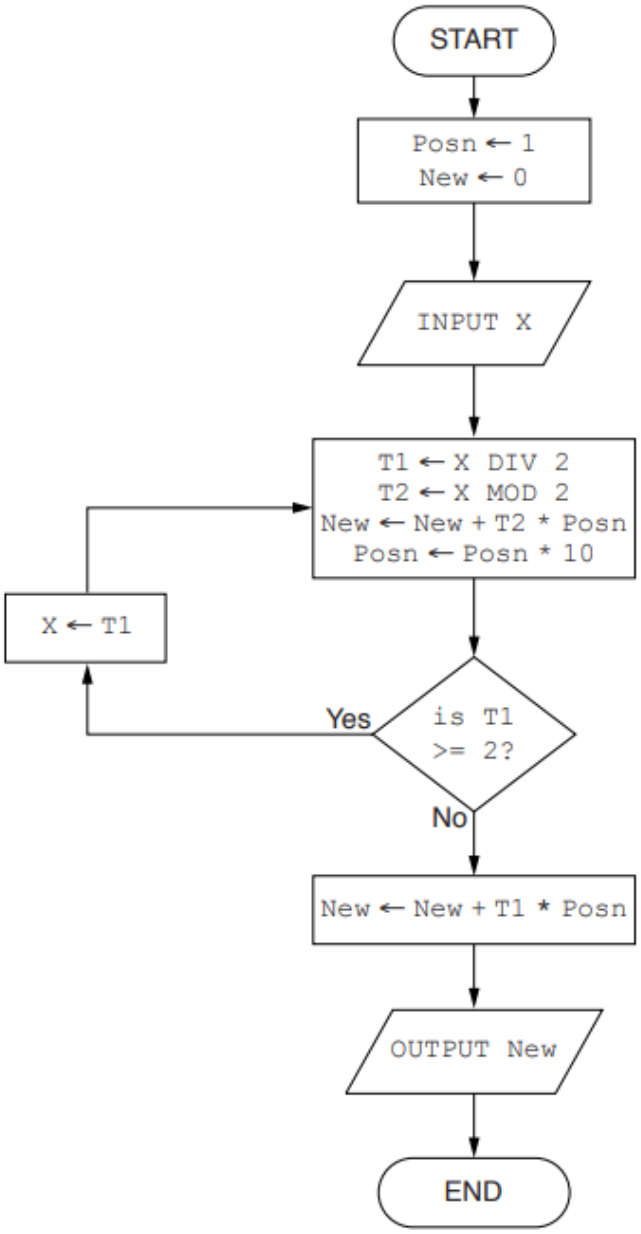
Complete the trace table for the input data:

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]

141 **(a)** The flowchart inputs an integer. The predefined function `DIV` gives the integer result of the division, e.g. `Y ← 10 DIV 3` gives the value `Y = 3`. The predefined function `MOD` gives the value of the remainder, e.g. `Y ← 10 MOD 3` gives the value `Y = 1`.



Complete a trace table for each of the **two** input values 5 and 12.

Trace table for input value 5

X	Posn	New	T1	T2	OUTPUT

Trace table for input value 12

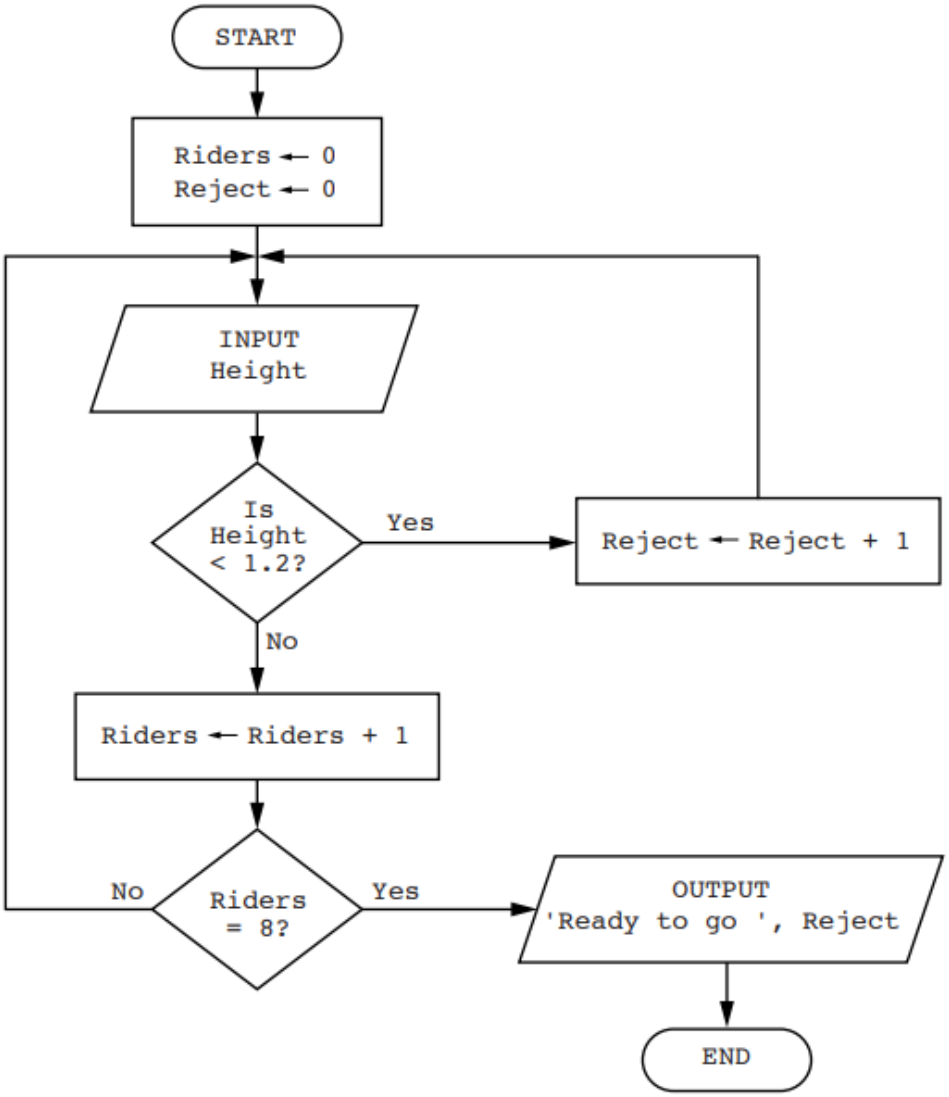
X	Posn	New	T1	T2	OUTPUT

[6]

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

.....[1]

142 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.



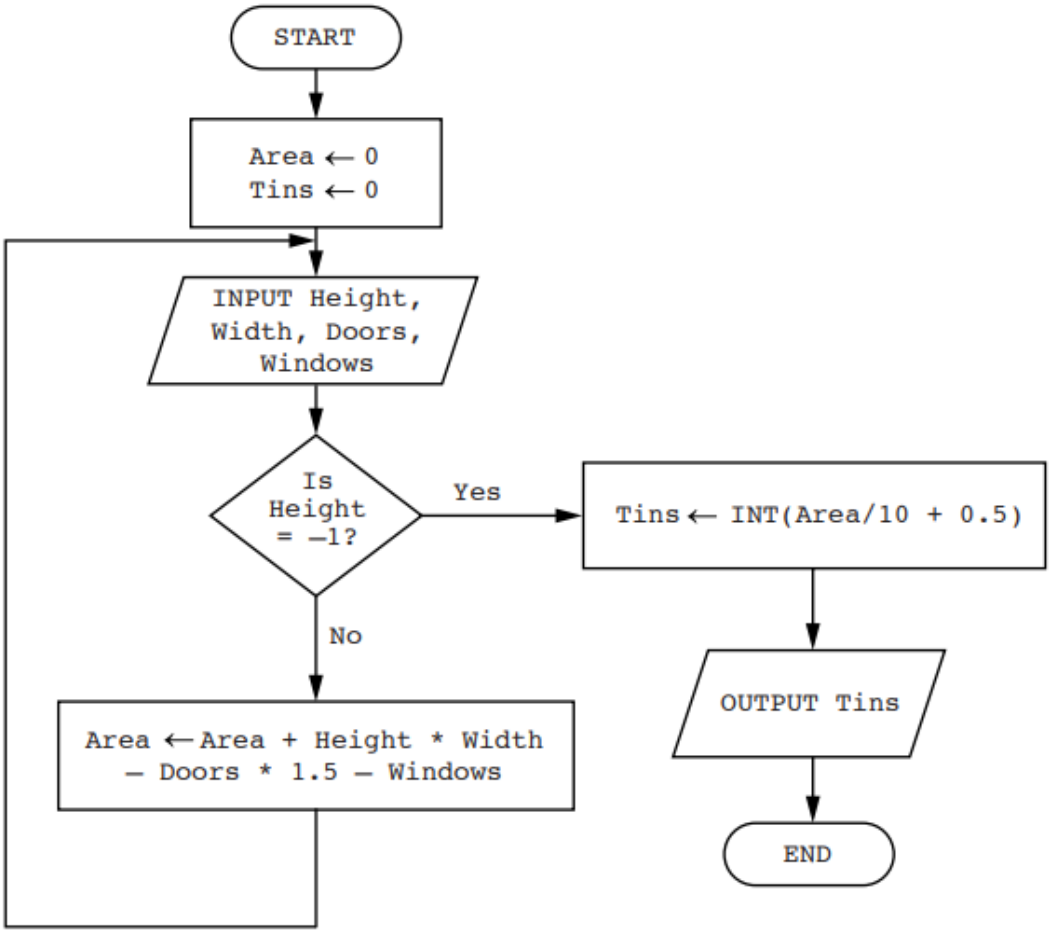
Complete the trace table for the input data:

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]

143 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.



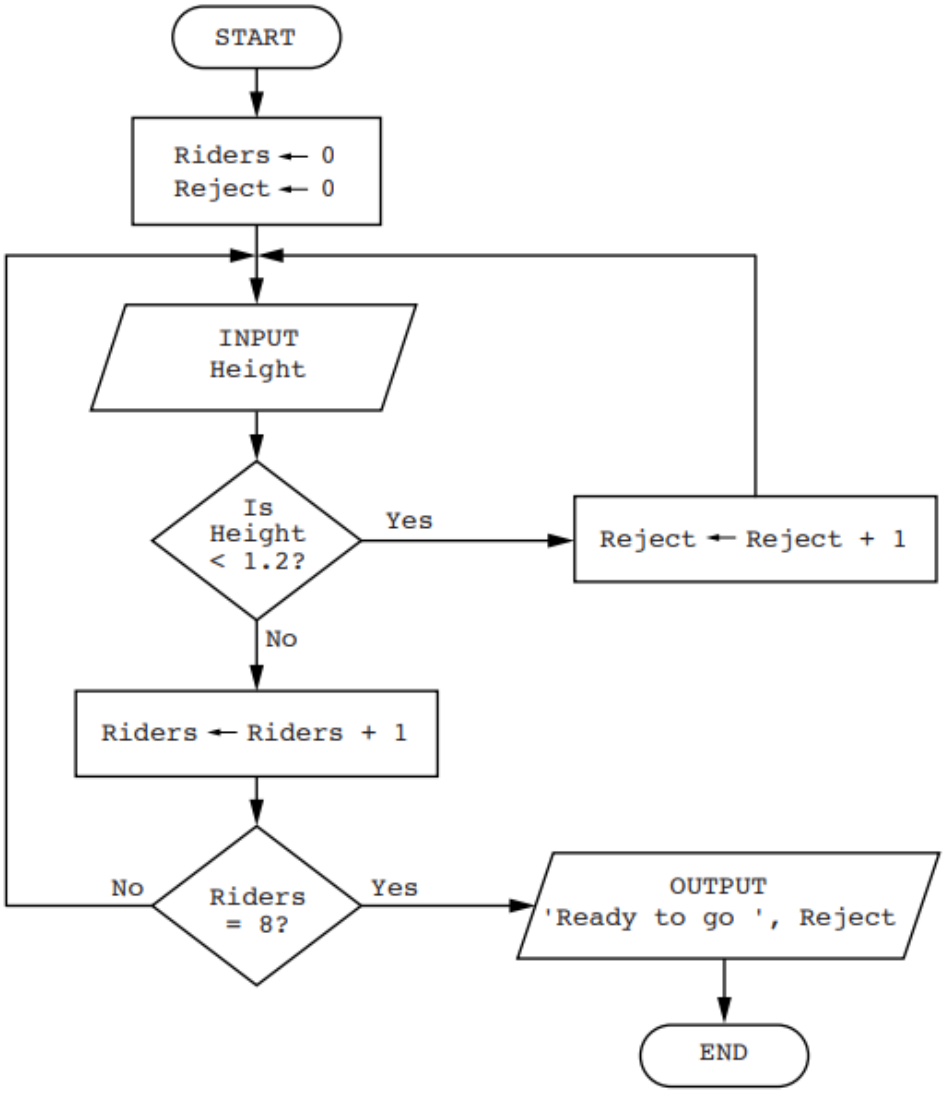
Complete the trace table for the input data:

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

[4]

144 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:

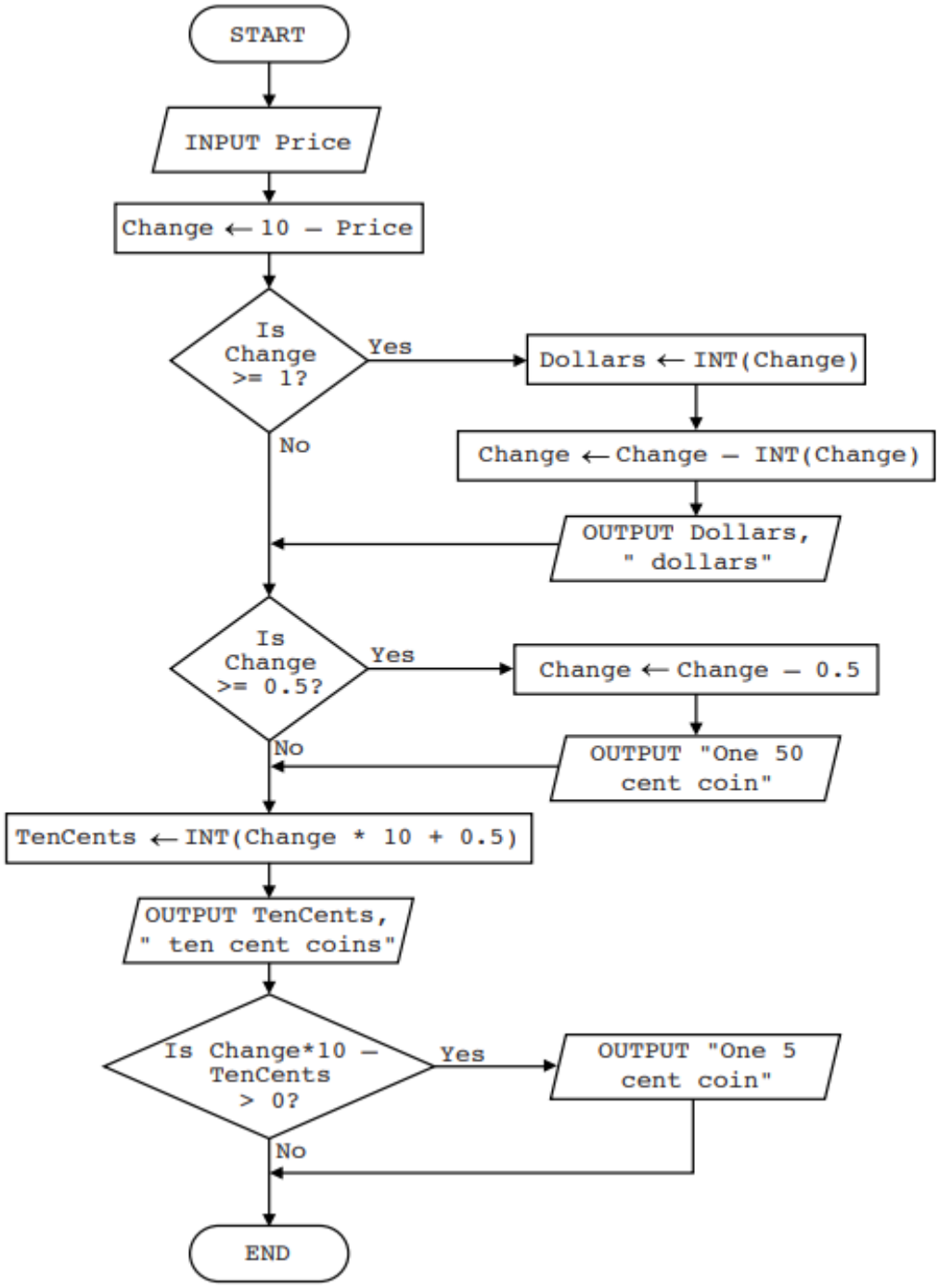
1.4, 1.3, 1.1, 1.3, 1.0, 1.5, 1.2, 1.3, 1.4, 1.3, 0.9, 1.5, 1.6, 1.0

Riders	Reject	Height	OUTPUT

[4]

145 The flowchart below inputs the price of an item under $10. The change from a $10 note is output. Any amount less than 5 cents is rounded up to 5 cents.

The predefined function `INT` rounds a number down to the nearest whole number; for example `Z ← INT(5.7)` gives the value `Z = 5`

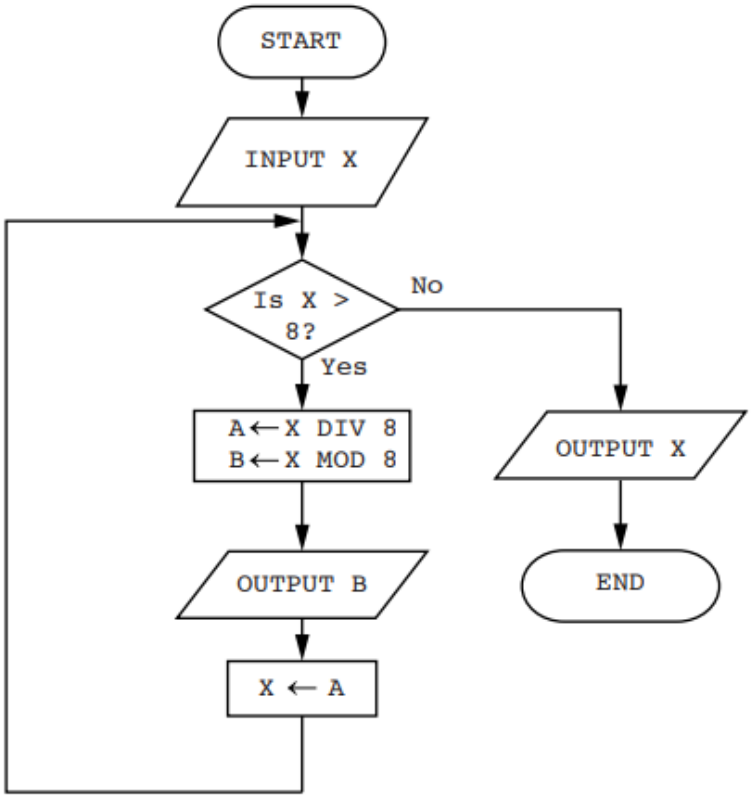


Complete the trace table for the input data: 6.29

Price	Change	Dollars	TenCents	OUTPUT

[5]

146 The flowchart below inputs an integer. The predefined function `DIV` gives the value of the division, for example `Z ← 11 DIV 3` gives the value `Z = 3`. The predefined function `MOD` gives the value of the remainder, for example `Z ← 11 MOD 3` gives the value `Z = 2`.



Complete a trace table for each of the two input values **33** and **75**.

Trace table for input value **33**

X	A	B	OUTPUT

Trace table for input value **75**

X	A	B	OUTPUT

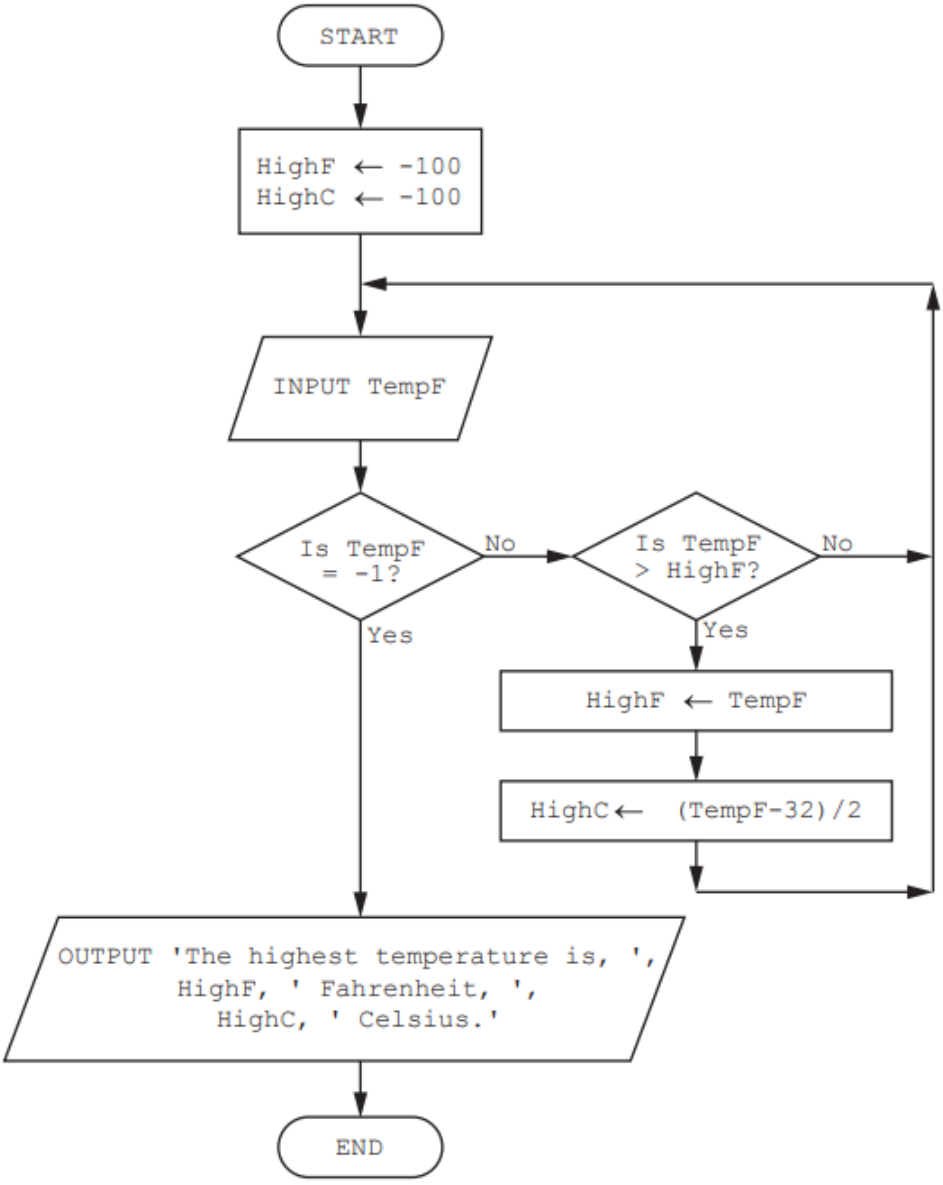
[4]

147 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.



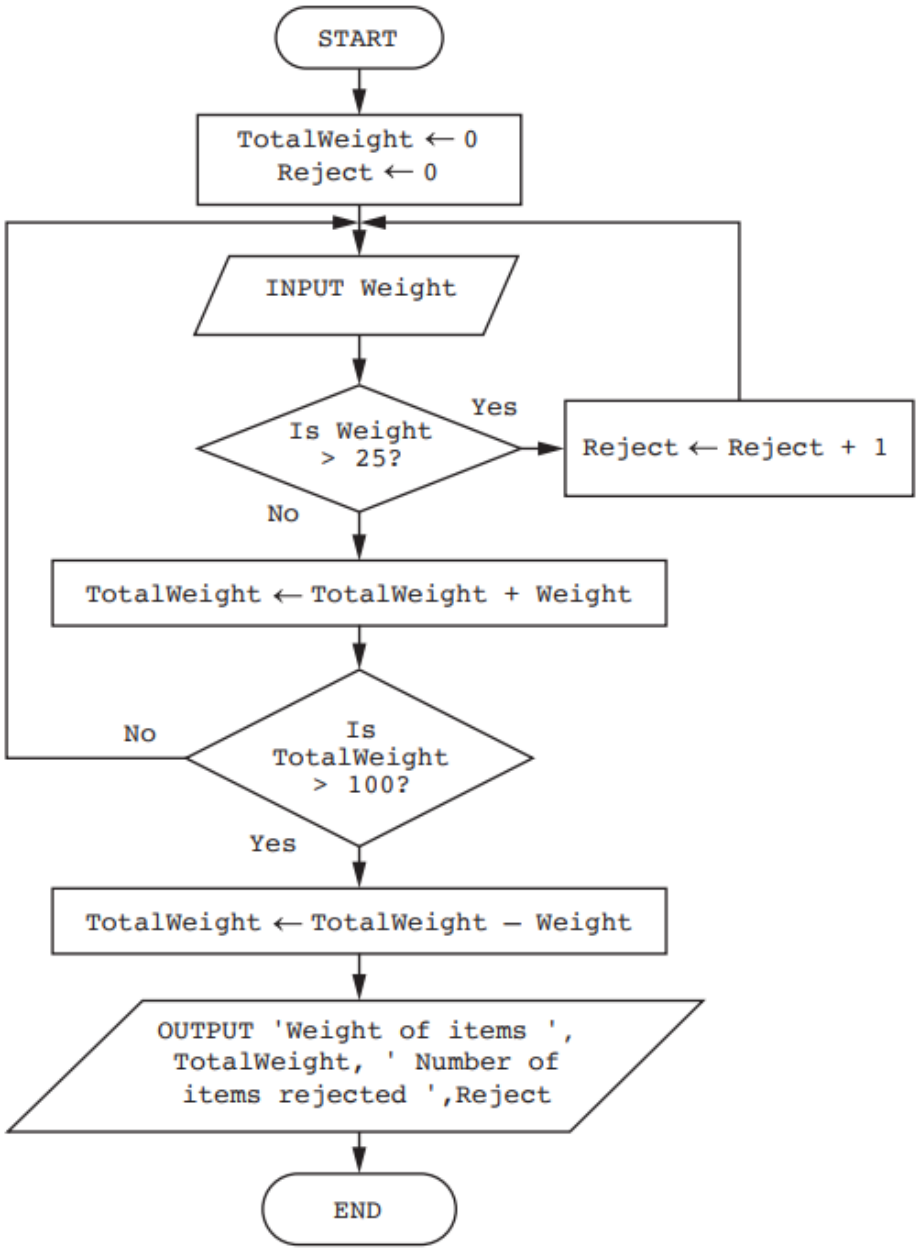
Complete the trace table for the input data:

68, 46, 50, 86, 65, 50, 40, 30, –1

HighF	HighC	TempF	OUTPUT

[5]

148 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.



Complete the trace table for the input data:

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

Weight	Reject	TotalWeight	OUTPUT

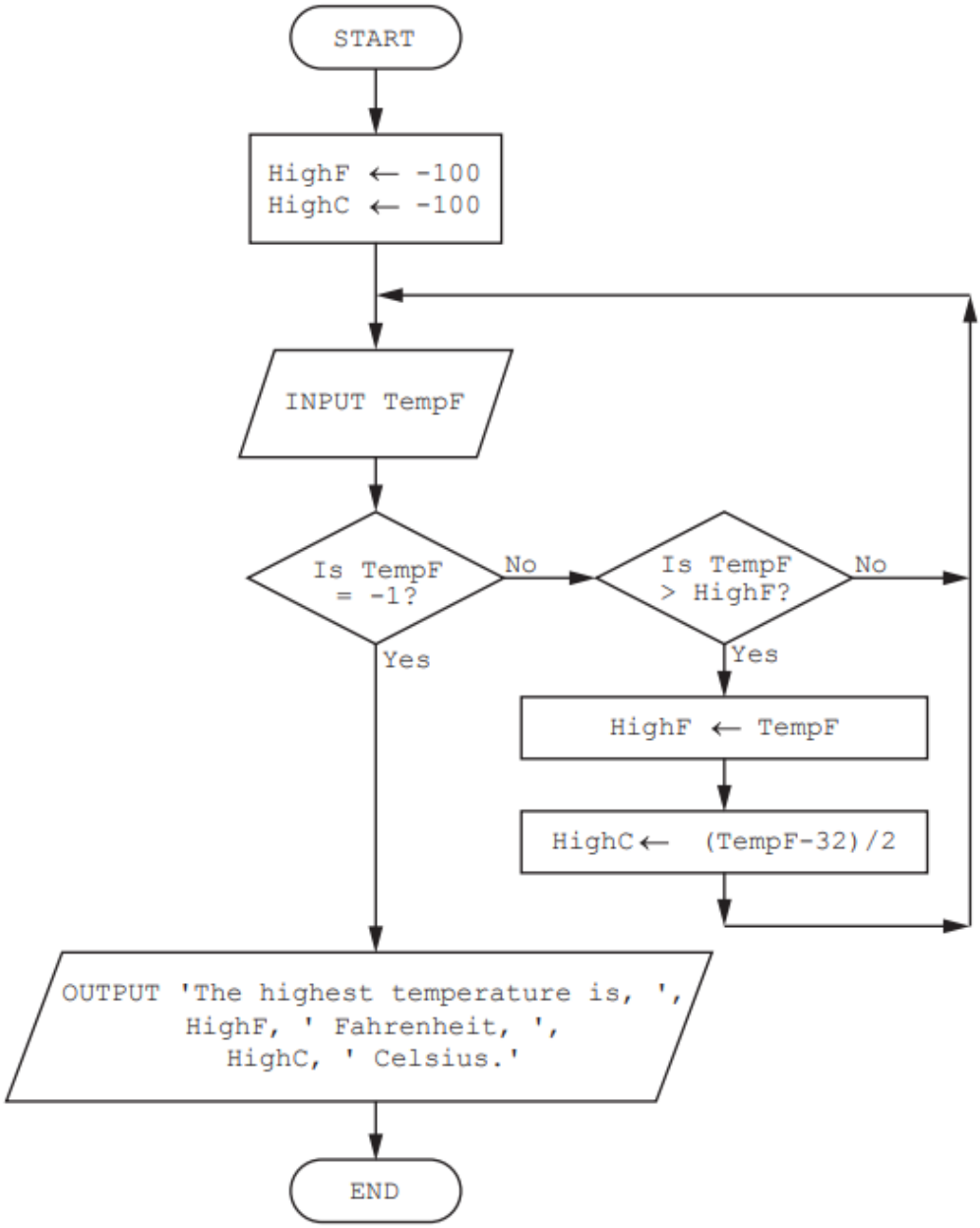
[5]

149 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.



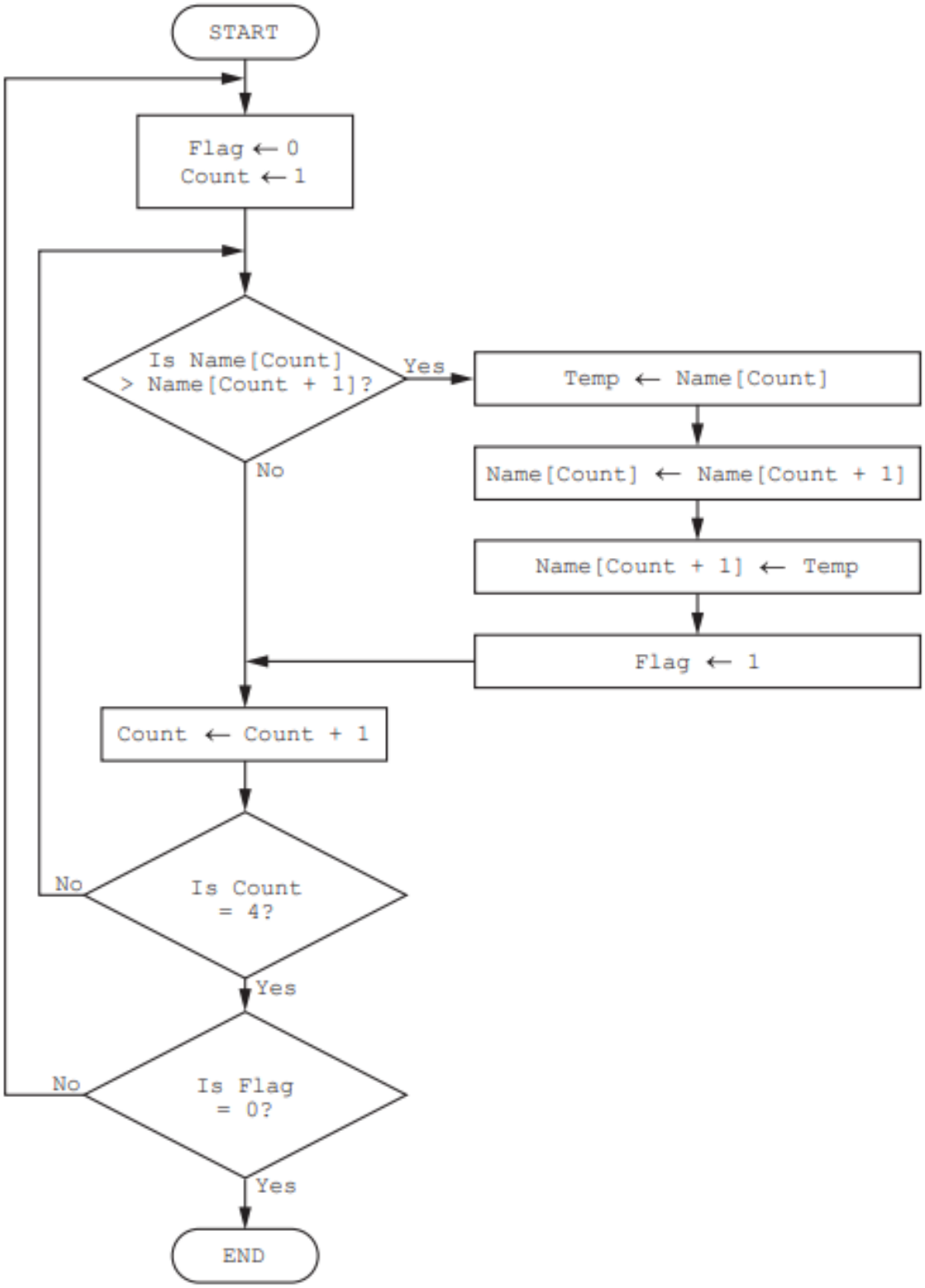
Complete the trace table for the input data:

68, 46, 50, 86, 65, 50, 40, 30, –1

HighF	HighC	TempF	OUTPUT

[5]

150 The flowchart below represents a program routine.



(a) The array used in the flowchart contains the following data:

Name[1]	Name[2]	Name[3]	Name[4]
Jamal	Amir	Eve	Tara

Complete the trace table using the data given in the array.

Flag	Count	Name[1]	Name[2]	Name[3]	Name[4]	Temp
		Jamal	Amir	Eve	Tara	

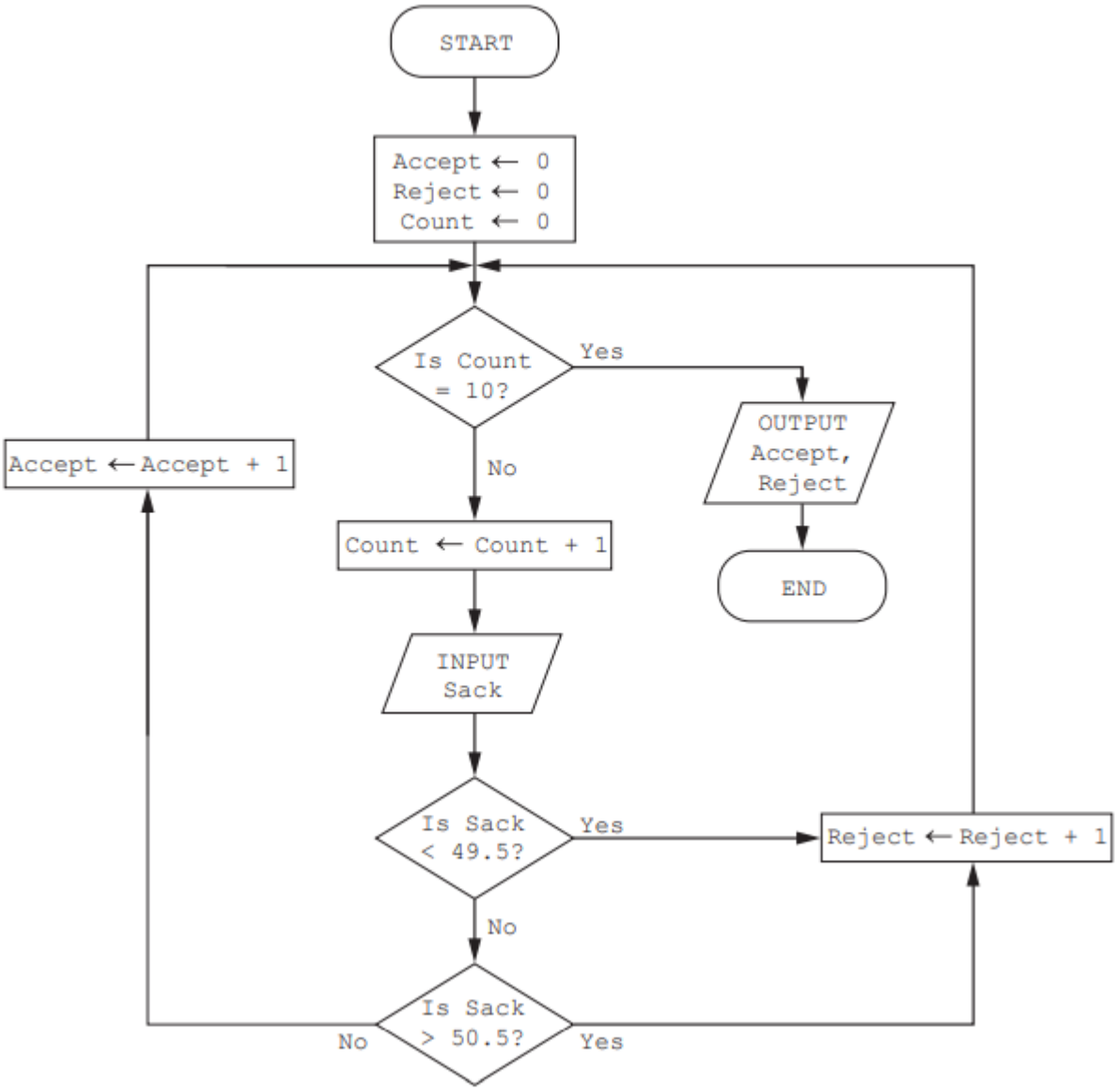
[5]

(b) Describe what the algorithm represented by the flowchart is doing.

.....

.....[2]

151 **(a)** This flowchart checks a batch of 10 rice sacks for weight. Sacks should weigh 50 kilograms each. Sacks weighing over 50.5 kilograms or less than 49.5 kilograms are rejected. The number of sacks accepted and the number of sacks rejected is output.



Complete the trace table for the input data:

50.4, 50.3, 49.1, 50.3, 50.0, 49.5, 50.2, 50.3, 50.5, 50.6

Accept	Reject	Count	Sack	OUTPUT

[5]

(b) The size of the batch has increased to 50 sacks. It has been decided to only reject sacks that are underweight.

State the changes that need to be made to the flowchart.

.....

.....

.....

.....[2]

152 **(a)** Draw a flowchart for an algorithm to input numbers. Reject any numbers that are negative and count how many numbers are positive. When the number zero is input, the process ends and the count of positive numbers is output.

(b) Explain the changes you will make to your algorithm to also count the negative numbers.

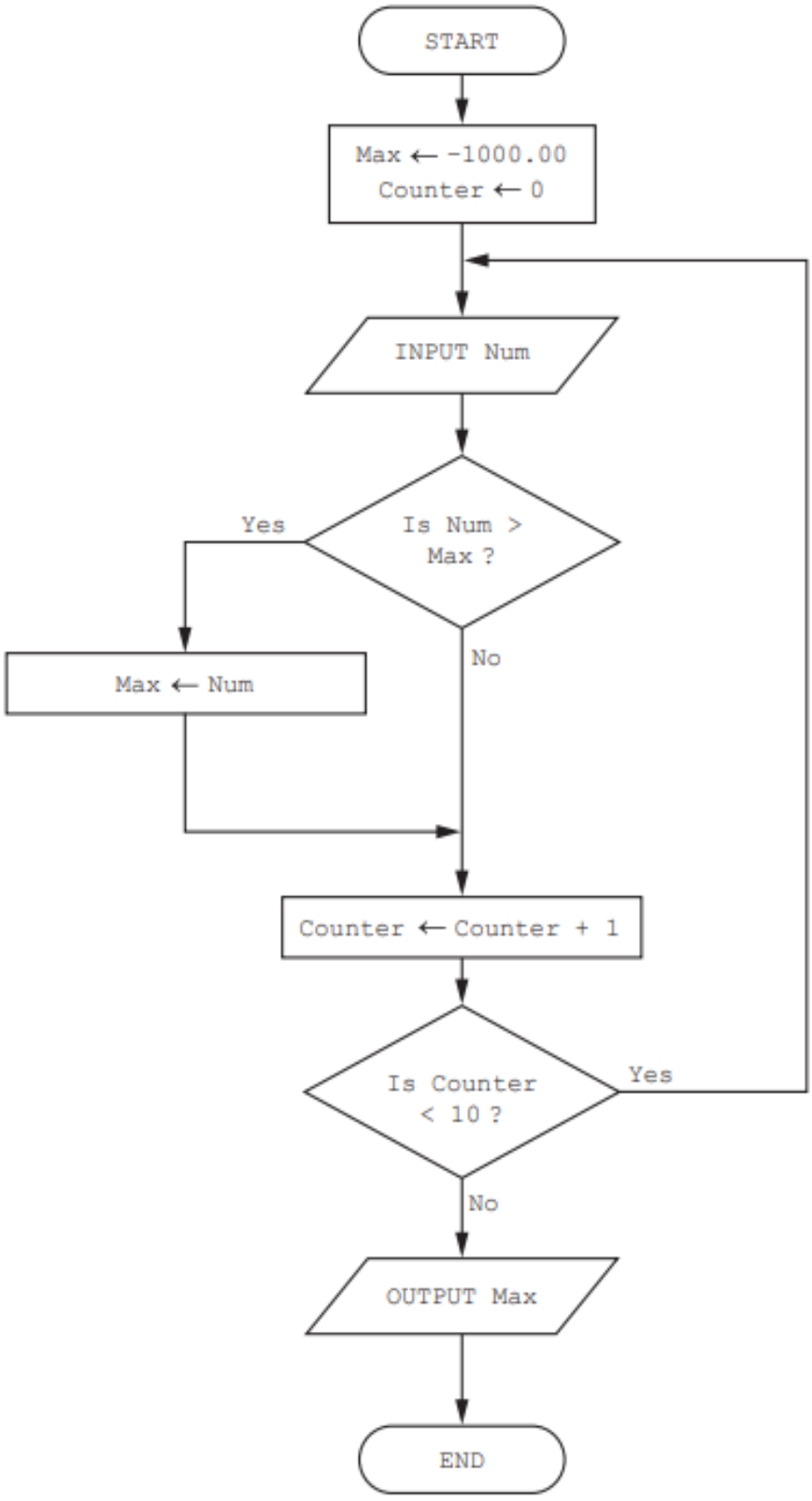
.....

.....

.....

.....[2]

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



(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]

(b) Describe **two** different changes you should make to the flowchart to find the smallest number instead of the largest number.

Change 1 .....

.....

Change 2 .....

.....

[2]

154 **Six** terms associated with programming and **six** descriptions are listed.

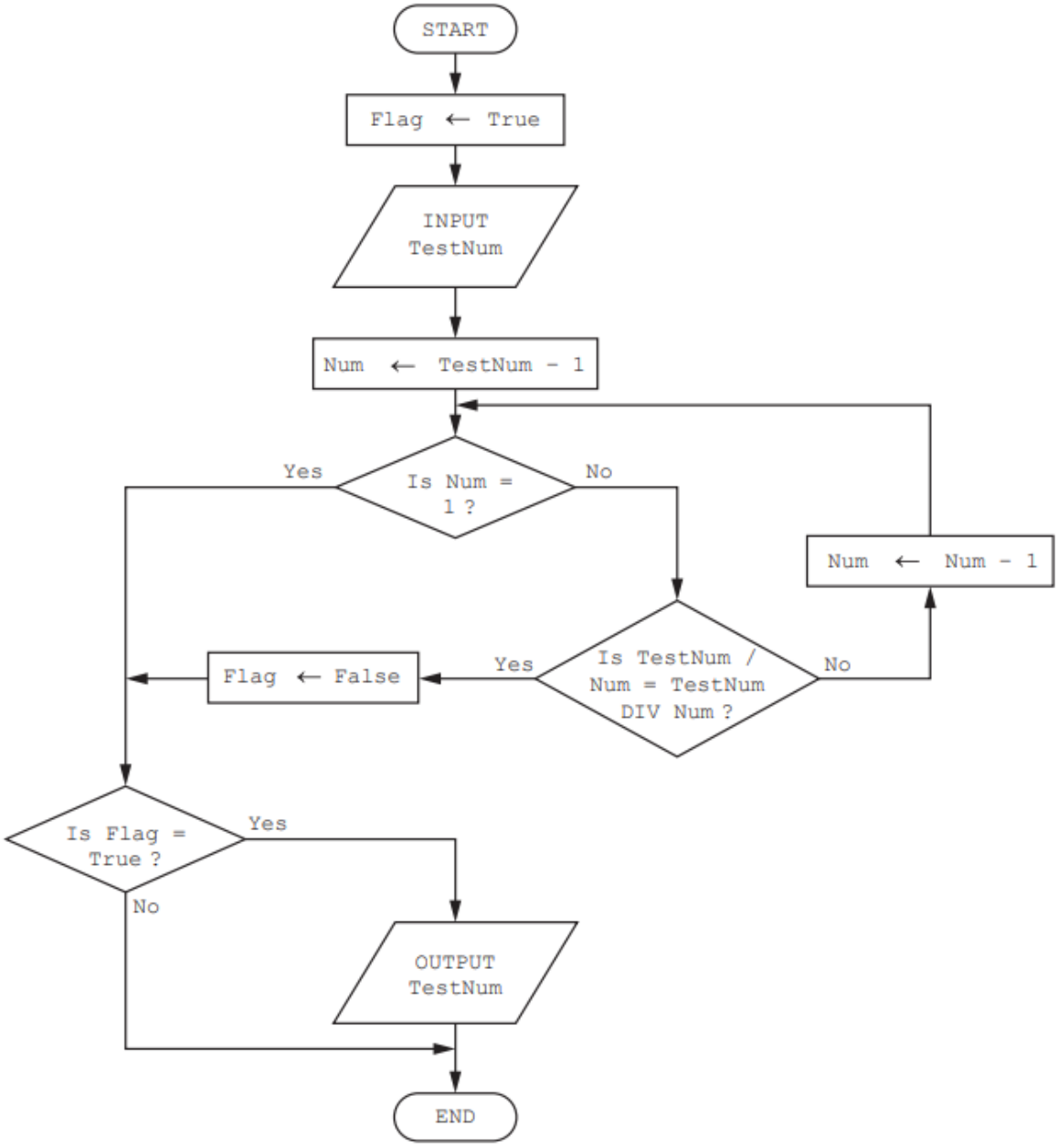
Draw a line to link each term with its most appropriate description.

Term	Description
Top-down design	Pre-written code to include in your own program to carry out a common task.
Structure diagram	Shows the steps representing an algorithm using various shapes of boxes.
Flowchart	Shows the hierarchy of the different components which make up a system.
Pseudocode	Shows the values of variables as you manually test your program.
Library routine	Breaks down a system into successively smaller pieces.
Trace table	Describes a program using a simplified high-level notation.

[5]

155 The flowchart performs a mathematical process on a number input called `TestNum`

`DIV` is used to represent **integer division** e.g. `7 DIV 3 = 2`



(a) Complete the trace table for the input data: **7**

Flag	TestNum	Num	OUTPUT

[2]

(b) Complete the trace table for the input data: **6**

Flag	TestNum	Num	OUTPUT

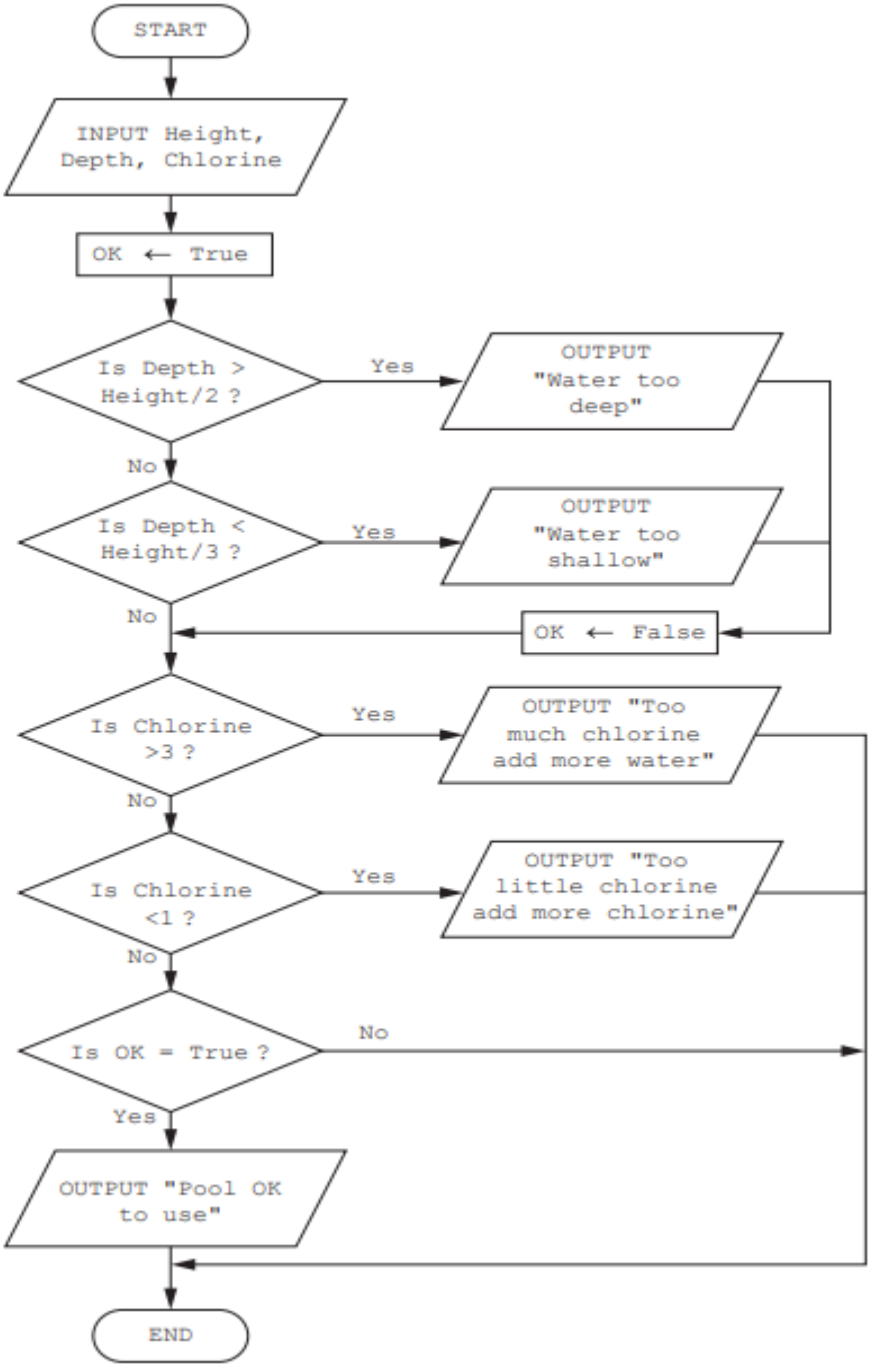
[2]

(c) State the purpose of the algorithm in the flowchart.

.....

.....[1]

156 The flowchart checks the level of chlorine and the depth of water compared to the height of the swimming pool. Error messages are output if a problem is found.



(a) Complete the trace tables for each set of input data.

Input data: **6**, **2.5**, **2**

Height	Depth	Chlorine	OK	OUTPUT

Input data: **4**, **3**, **1.5**

Height	Depth	Chlorine	OK	OUTPUT

Input data: **6**, **3.5**, **4**

Height	Depth	Chlorine	OK	OUTPUT

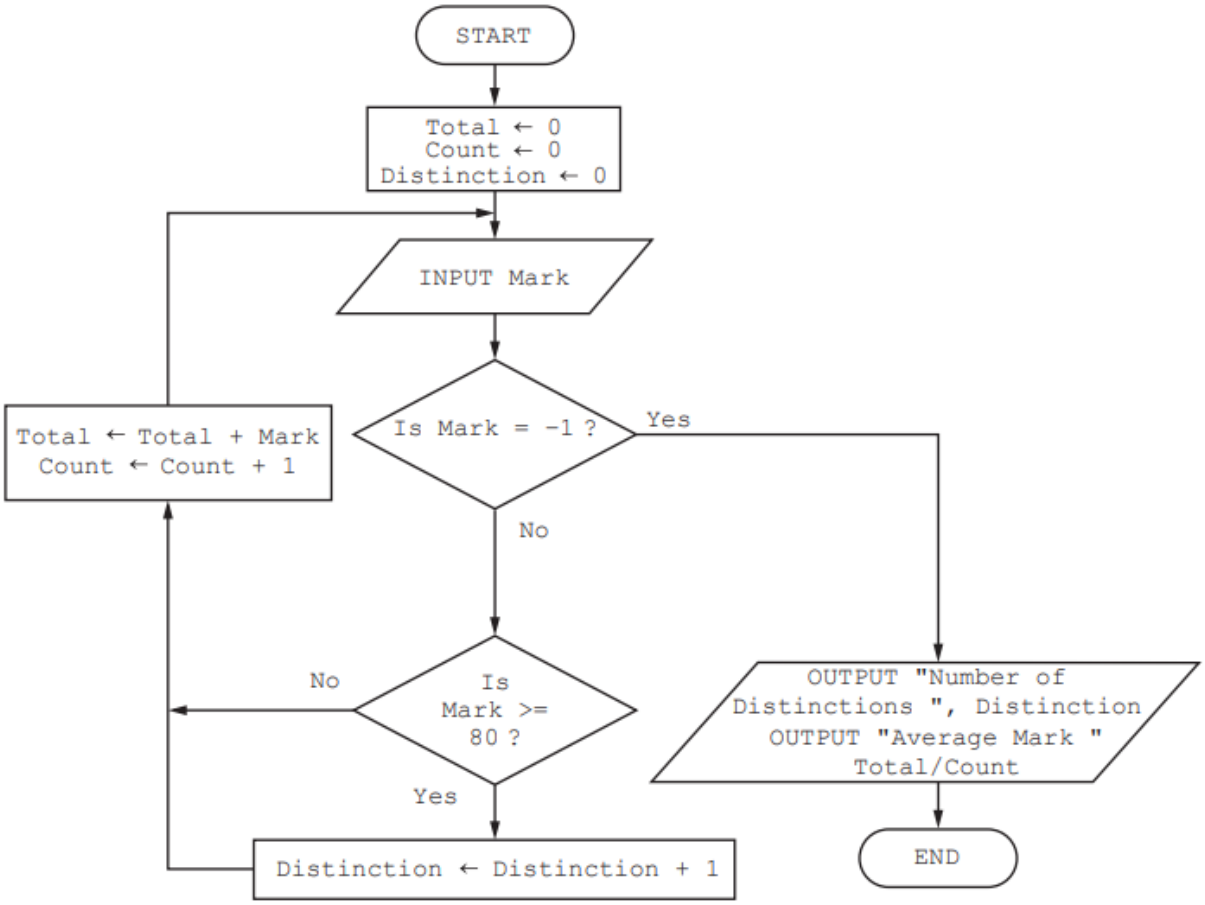
[6]

(b) Identify a problem with the algorithm that the flowchart represents.

.....

..... [1]

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

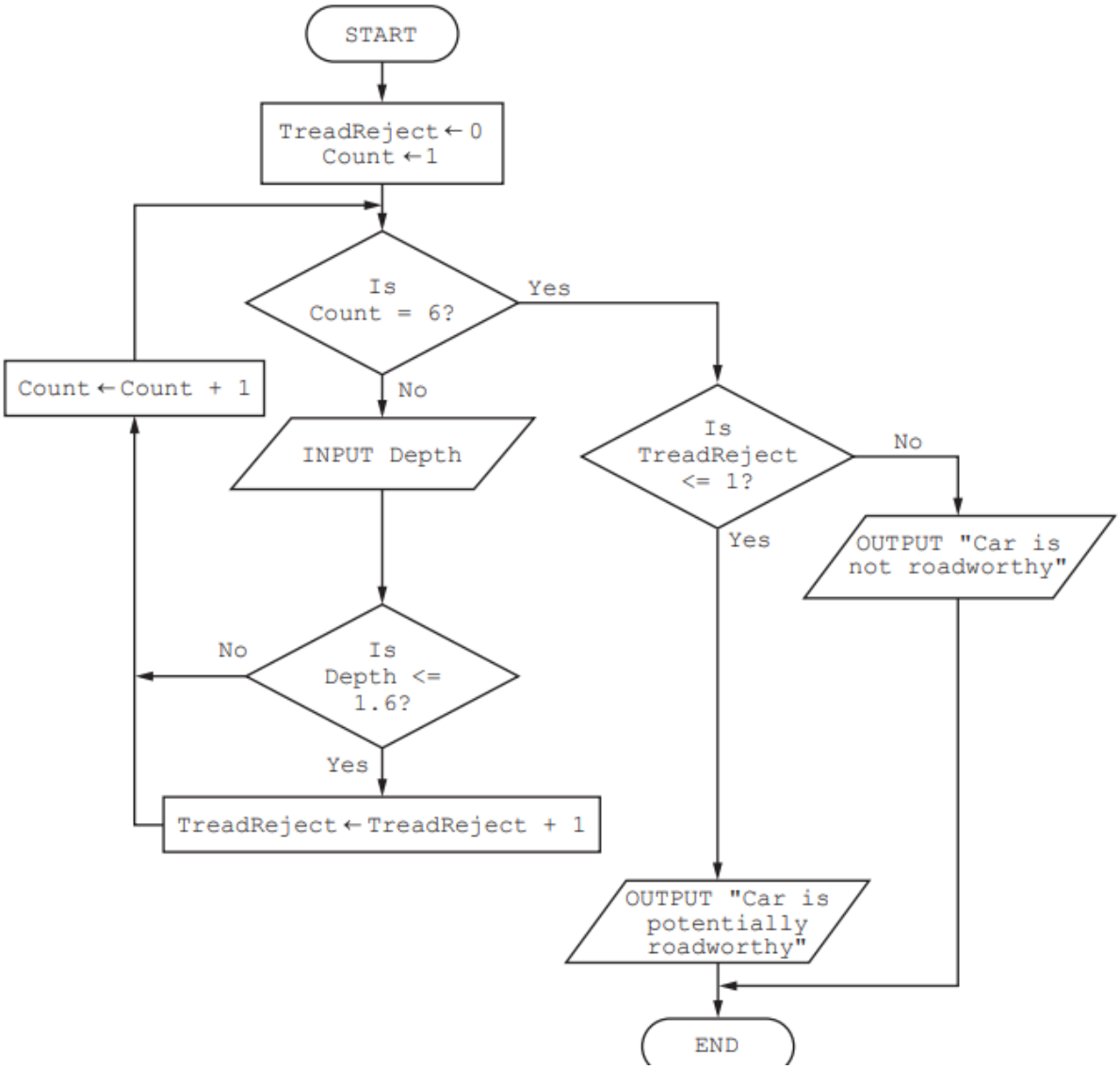


Complete the trace table for the mark input data: 50, 70, 65, 30, 95, 50, 55, 85, 65, 35, –1, 45

Total	Count	Distinction	Mark	**OUTPUT**

[4]

158 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.



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]

159 For each of the **four** descriptions in the table, place a tick in the correct column to show whether it describes a **Structure diagram**, a **Flowchart** or **Library routines**.

Description	Structure diagram	Flowchart	Library routines
A modelling tool used to show the hierarchy of a system.			
A collection of standard programs available for immediate use.			
A graphical representation used to represent an algorithm.			
A graphical representation to show how a system is broken into sub-systems.			

[4]

160 Draw **four** different flowchart symbols and describe how they are used in a program flowchart.

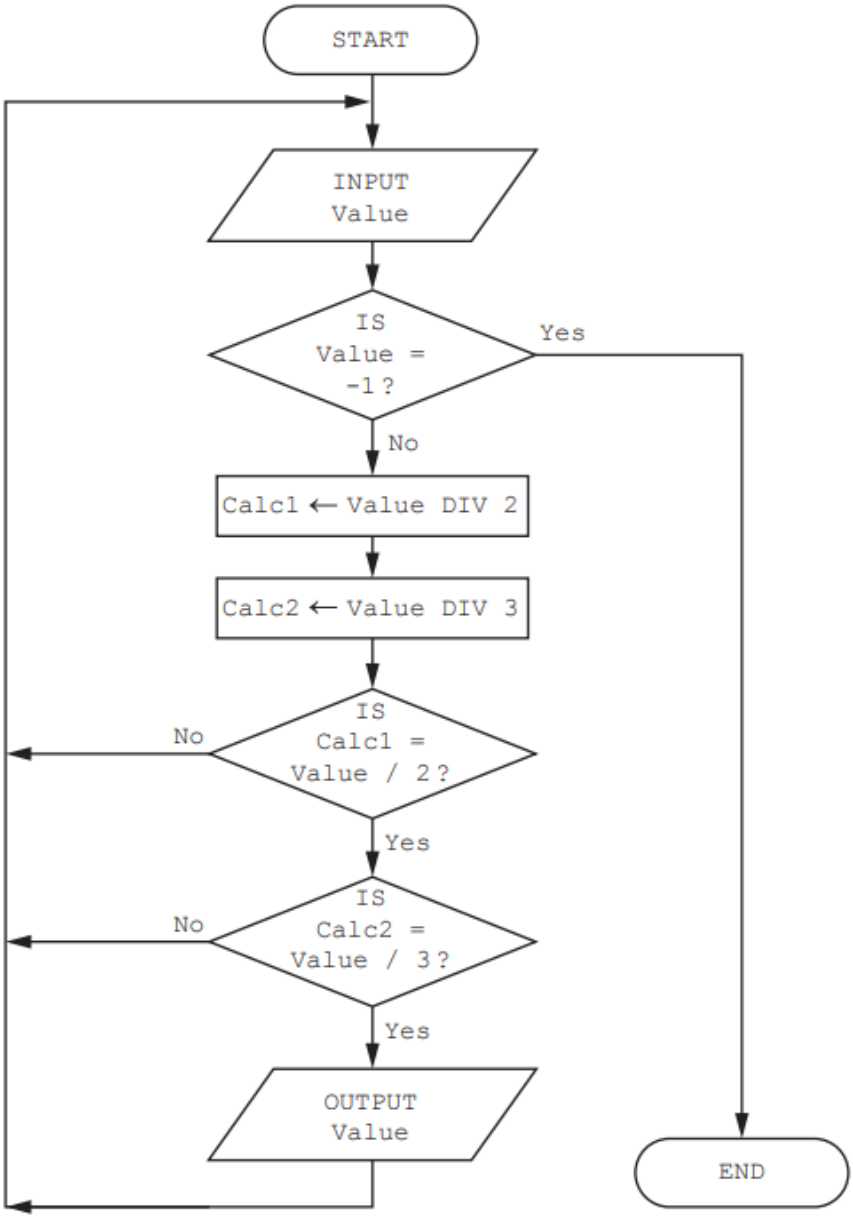
Flowchart symbol	Description of use

[4]

161 The flowchart represents an algorithm.

The predefined function `DIV` gives the value of the result of integer division, for example, `y ← 9 DIV 4` gives `y` a value of `2`

An input value of –1 ends the algorithm.



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

50, 33, 18, 15, 30, –1, 45, 12, 90, 6

Value	Calc1	Calc2	OUTPUT

[4]

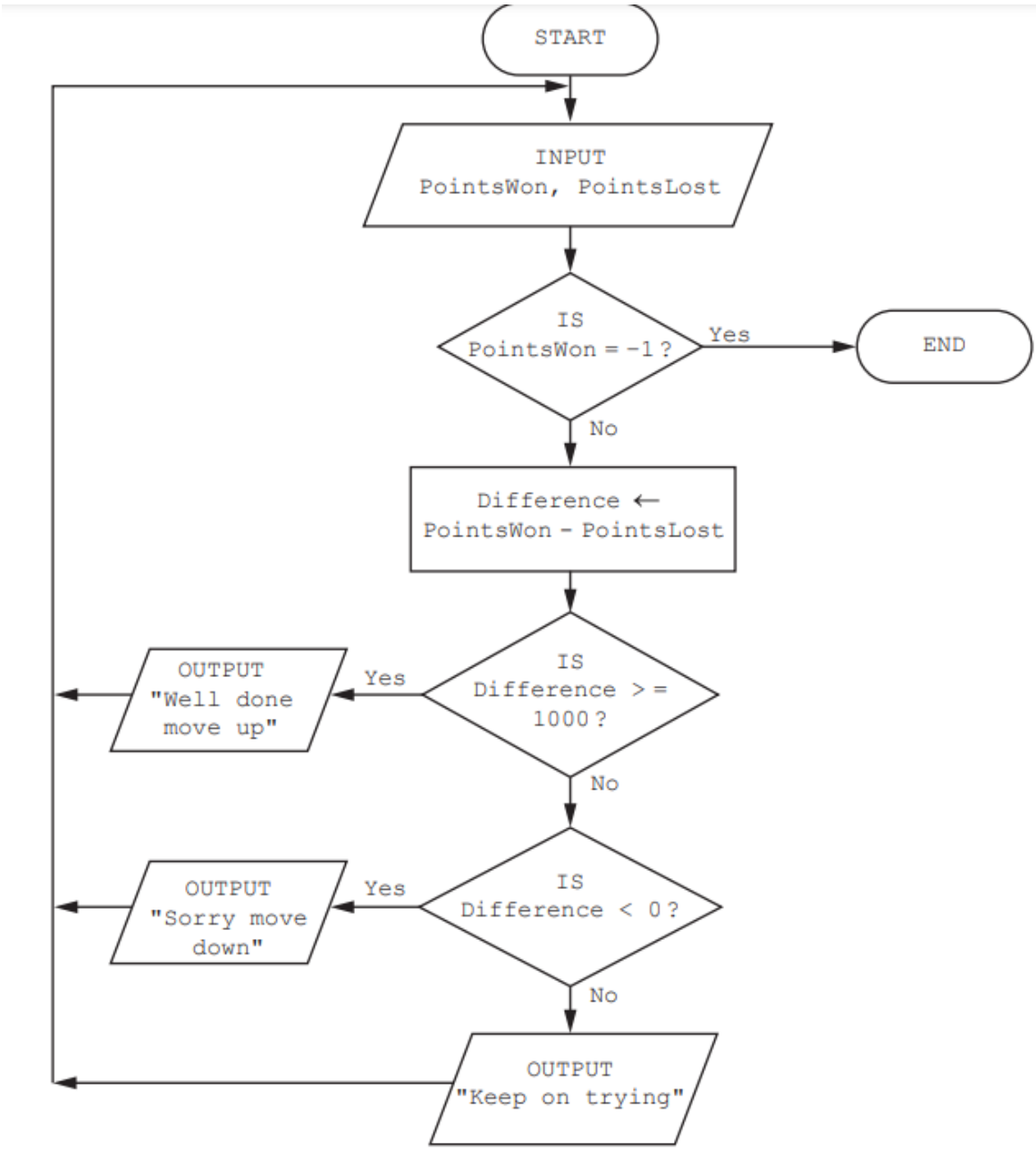
(b) Describe the purpose of the algorithm.

.....

.....

..... [2]

162 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]

(b) The flowchart needs to be changed. When the difference is more than 5000 the output message is 'Fantastic leap up two levels'.

Describe the changes that will need to be made to the flowchart.

.....

.....

.....

.....

.....

.....

.....

..... [3]

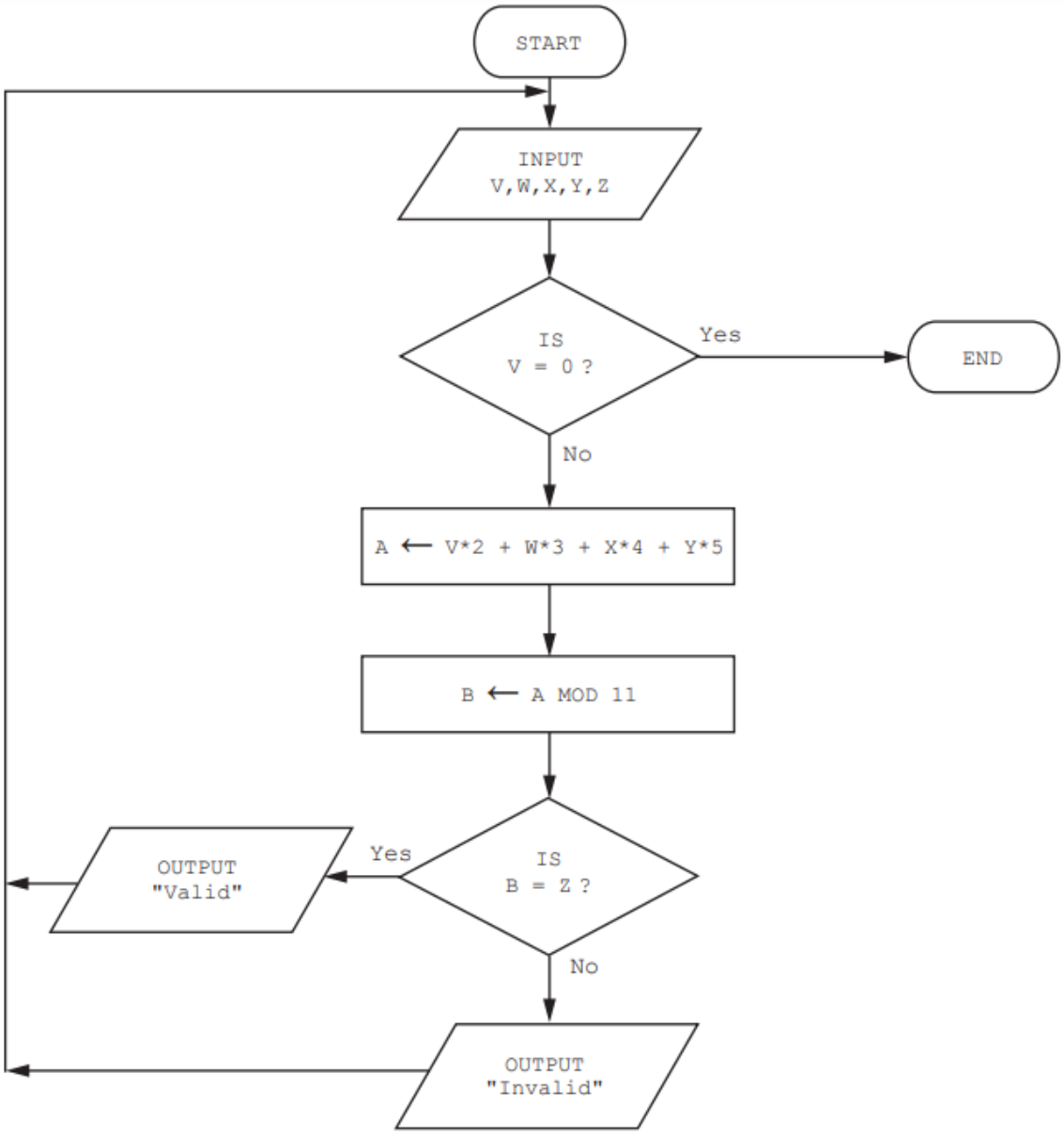
163 Draw the flowchart symbol for **Decision** and the flowchart symbol for **Process**.

Decision	Process

[2]

164 This flowchart inputs five numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example
`R ← 25 MOD 11` gives `R` a value of `3`



(a) Complete the trace table for this set of input data:
5, 4, 6, 2, 1, 9, 3, 2, 1, 6, 7, 6, 1, 5, 1, 0, 0, 0, 0, 0

V	W	X	Y	Z	A	B	OUTPUT

[4]

(b) Describe the purpose of this flowchart.

.....

.....

.....

..... [2]

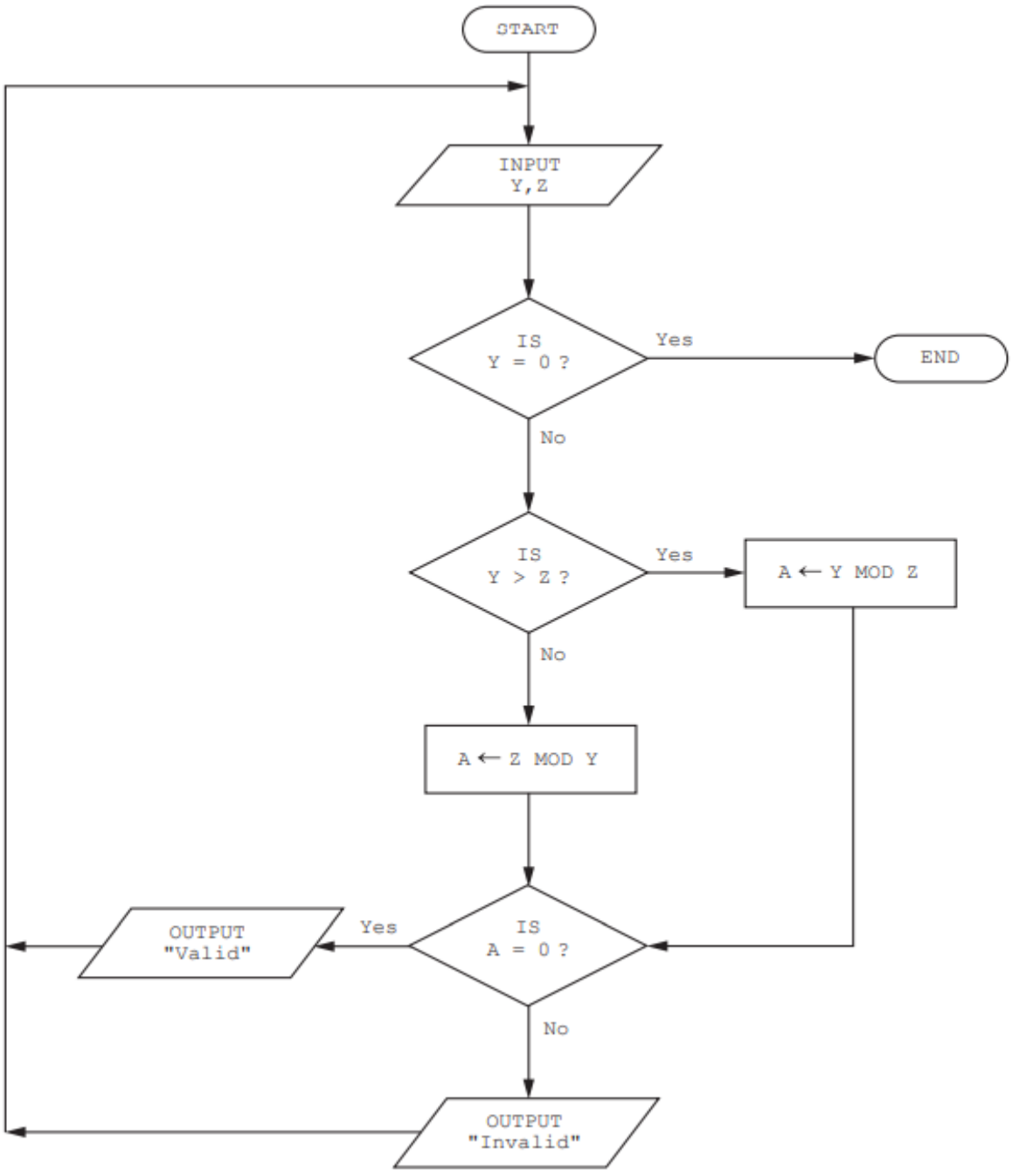
165 Draw a flowchart symbol to represent each of the following:

Input/Output	Decision

[2]

166 This flowchart represents an algorithm that allows the input of two numbers and performs a calculation.

The predefined function MOD finds the remainder from integer division for example `X ← 8 MOD 5` gives `X` a value of `3`.



(a) Complete a trace table for this set of input data:
11, 4, 6, 2, 3, 9, 3, 2, 2, 6, 0, 0, 1, 1

Y	Z	A	OUTPUT

[4]

(b) Explain the purpose of this algorithm.

.....

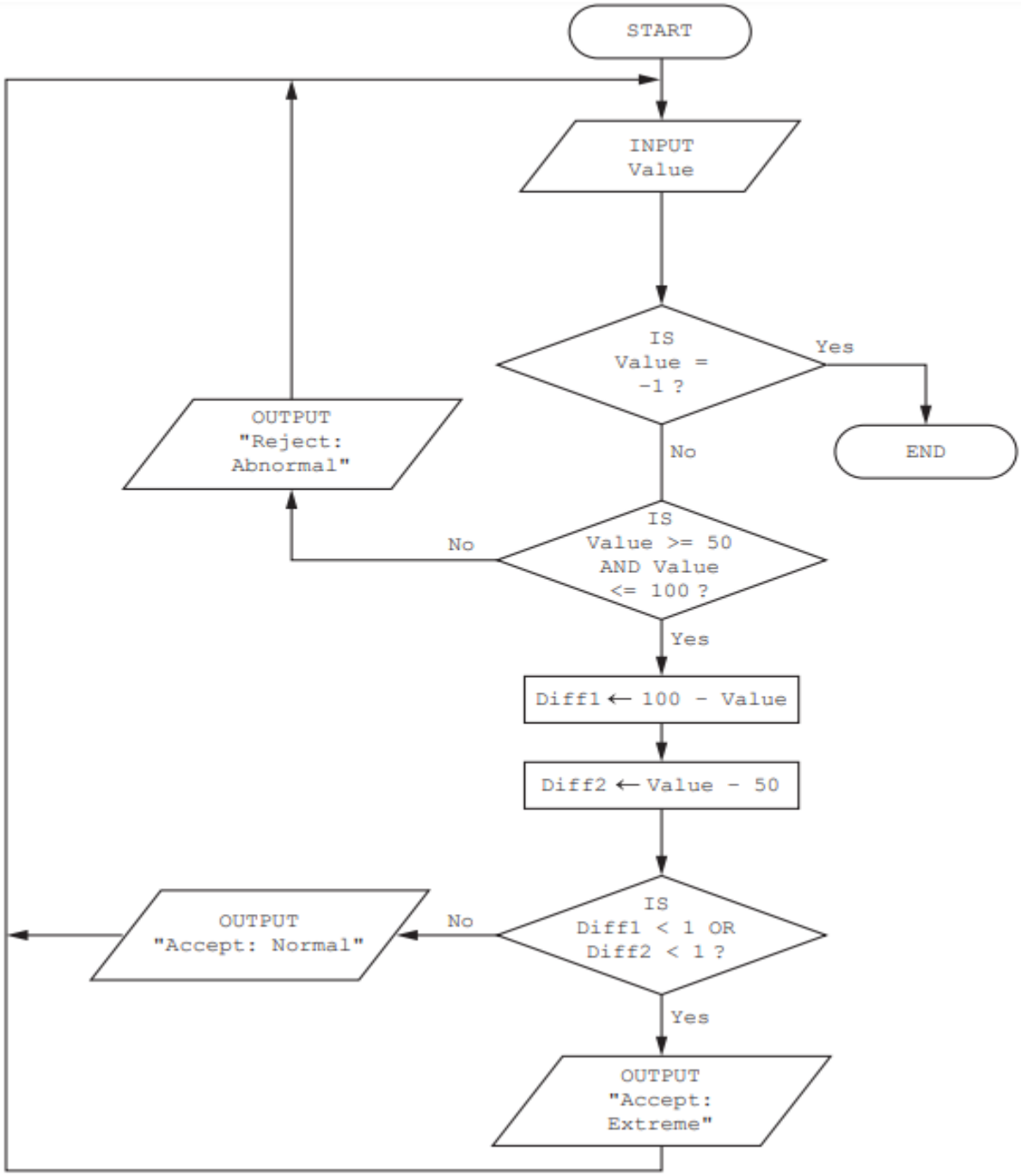
.....

.....

..... [2]

167 The flowchart represents an algorithm.

The algorithm will terminate if –1 is entered.



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

50, 75, 99, 28, 82, 150, –1, 672, 80

Value	Diff1	Diff2	OUTPUT

[4]

(b) Describe the purpose of the algorithm.

.....

.....

.....

..... [2]

168 **(a)** Draw the most appropriate flowchart symbol for each pseudocode statement.

Pseudocode statement	Flowchart symbol
`IF Number = 20`	
`PRINT Number`	
`Number ← Number + 1`	

[3]

(b) State the type of each pseudocode statement. For example, `X ← X + Y` is totalling.

`IF Number = 20` .....

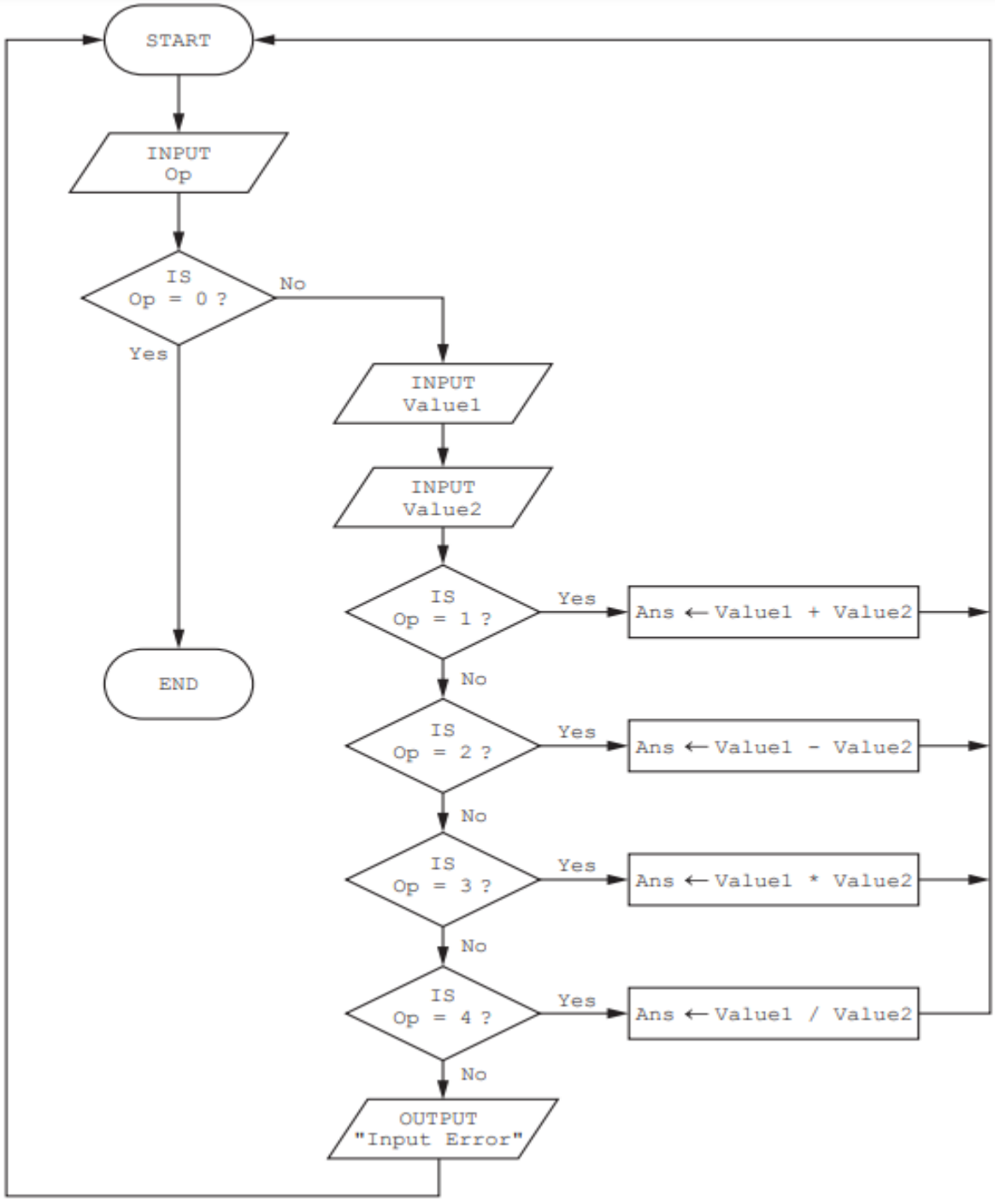
`PRINT Number` .....

`Number ← Number + 1` .....

[3]

169 The flowchart represents an algorithm.

The algorithm will terminate if 0 is entered at the `Op` input.



(a) Complete the trace table for the algorithm using this input data:

1, 87, 14, 3, 2, 30, 5, 10, 6, 4, 10, 2, 0, 2, 90, 6

Op	Value1	Value2	Ans	OUTPUT

[5]

(b) State the purpose of the algorithm.

.....

.....

.....

..... [1]

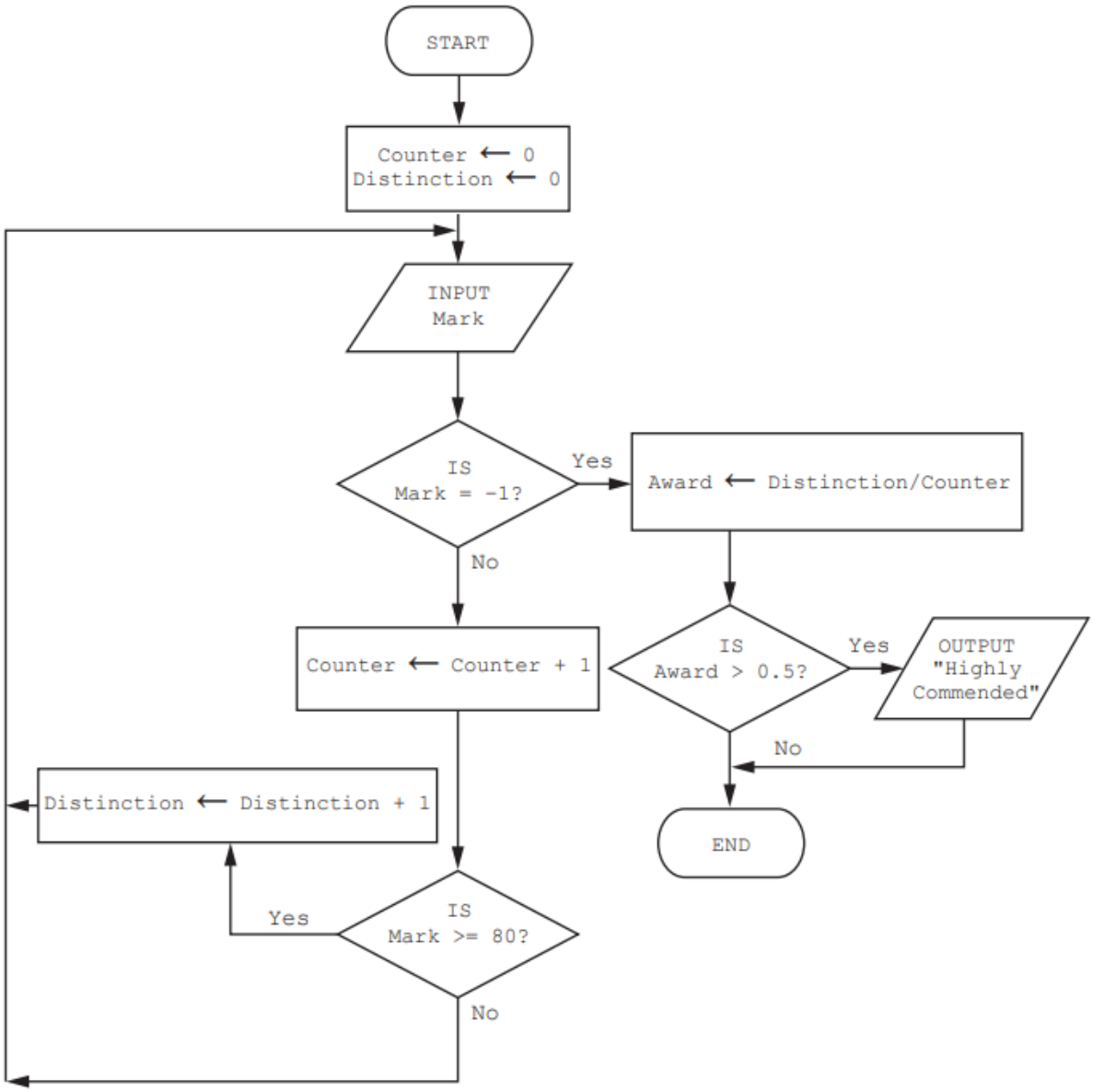
(c) Suggest an addition that could be made to the algorithm to make it more useful.

.....

.....

..... [1]

170 The algorithm shown by this flowchart allows the input of examination marks for a class of students. A mark of –1 ends the process. If a mark is 80 or over then a distinction grade is awarded. The number of distinctions for the whole class is calculated. If this is over 50% of the class, the class is awarded a highly commended certificate.



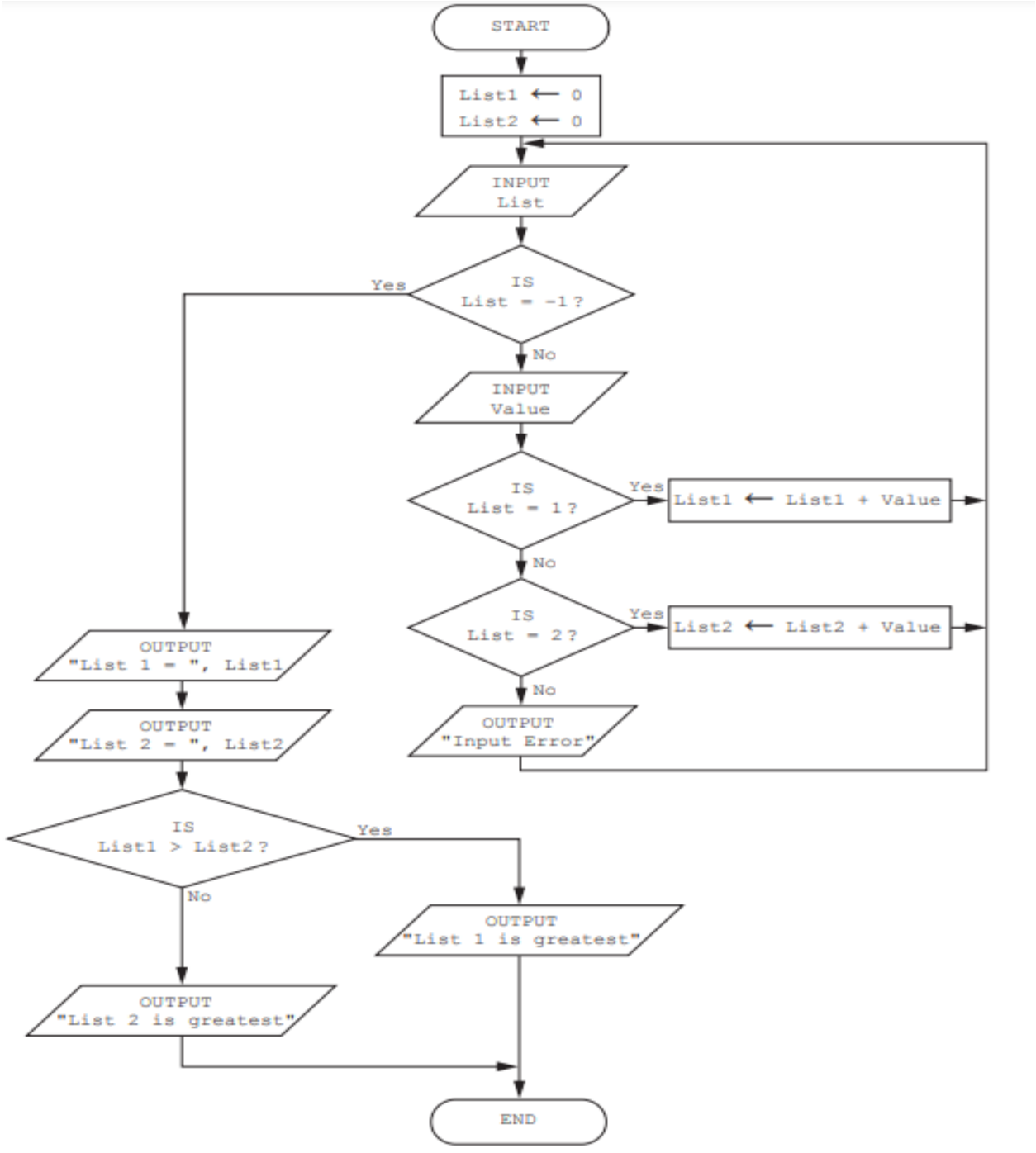
Complete a trace table for the algorithm using this input data:
88, 74, 60, 90, 84, 87, 95, 72, 84, 66, –1

Counter	Distinction	Mark	Award	OUTPUT

[5]

171 The flowchart represents an algorithm.

The algorithm will terminate if –1 is entered at the `List` input.



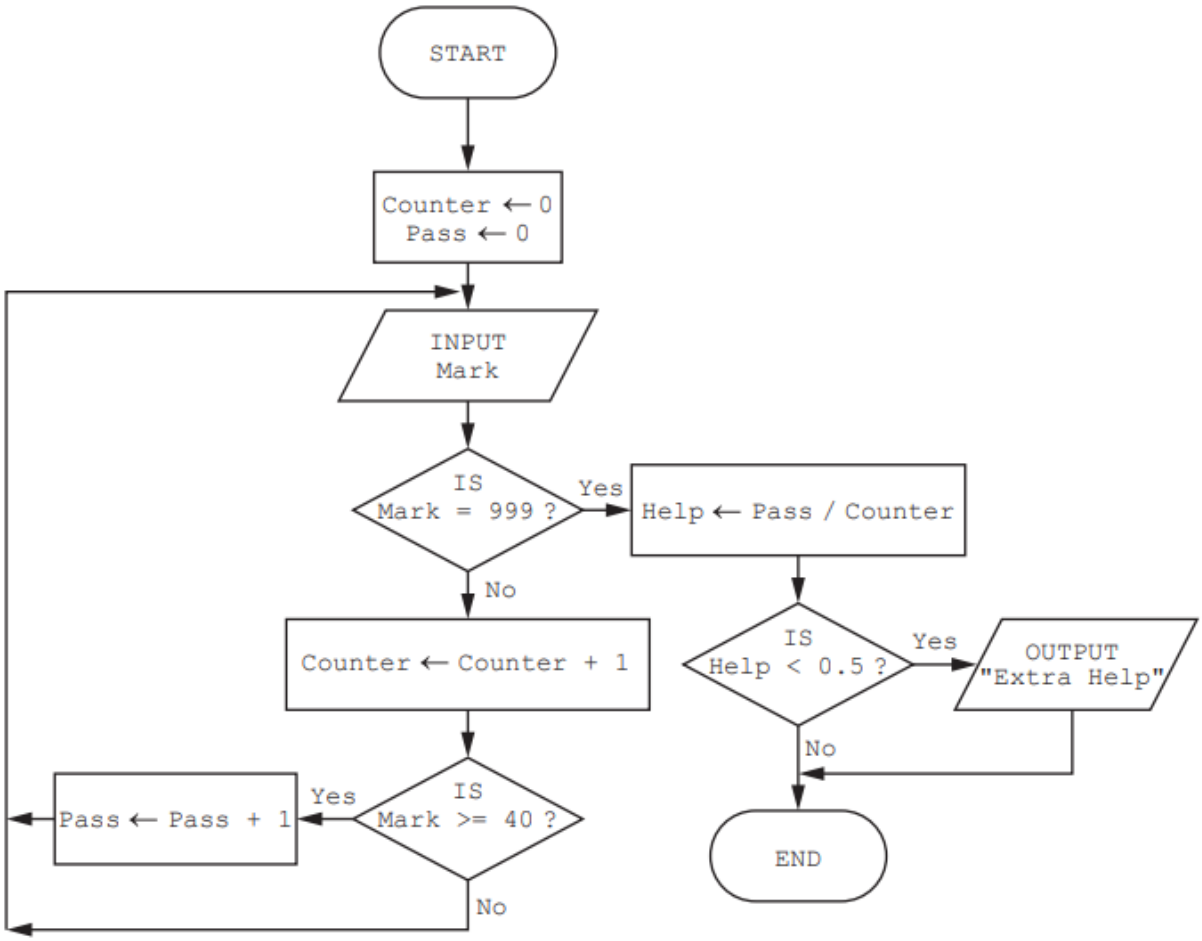
Complete the trace table for the algorithm using this input data:

2, 77, 2, 16, 1, 35, 2, –7, 5, 18, 1, 11, 1, 12, 2, 20, –1, 18

List	Value	List1	List2	OUTPUT

[5]

172 The algorithm, shown by this flowchart, allows the input of examination marks for a class of students. A mark of 999 ends the process. If a mark is 40 or over then a pass grade is awarded. The number of pass grades is calculated for the whole class. If this is under 50% of the class, the class is offered extra help.



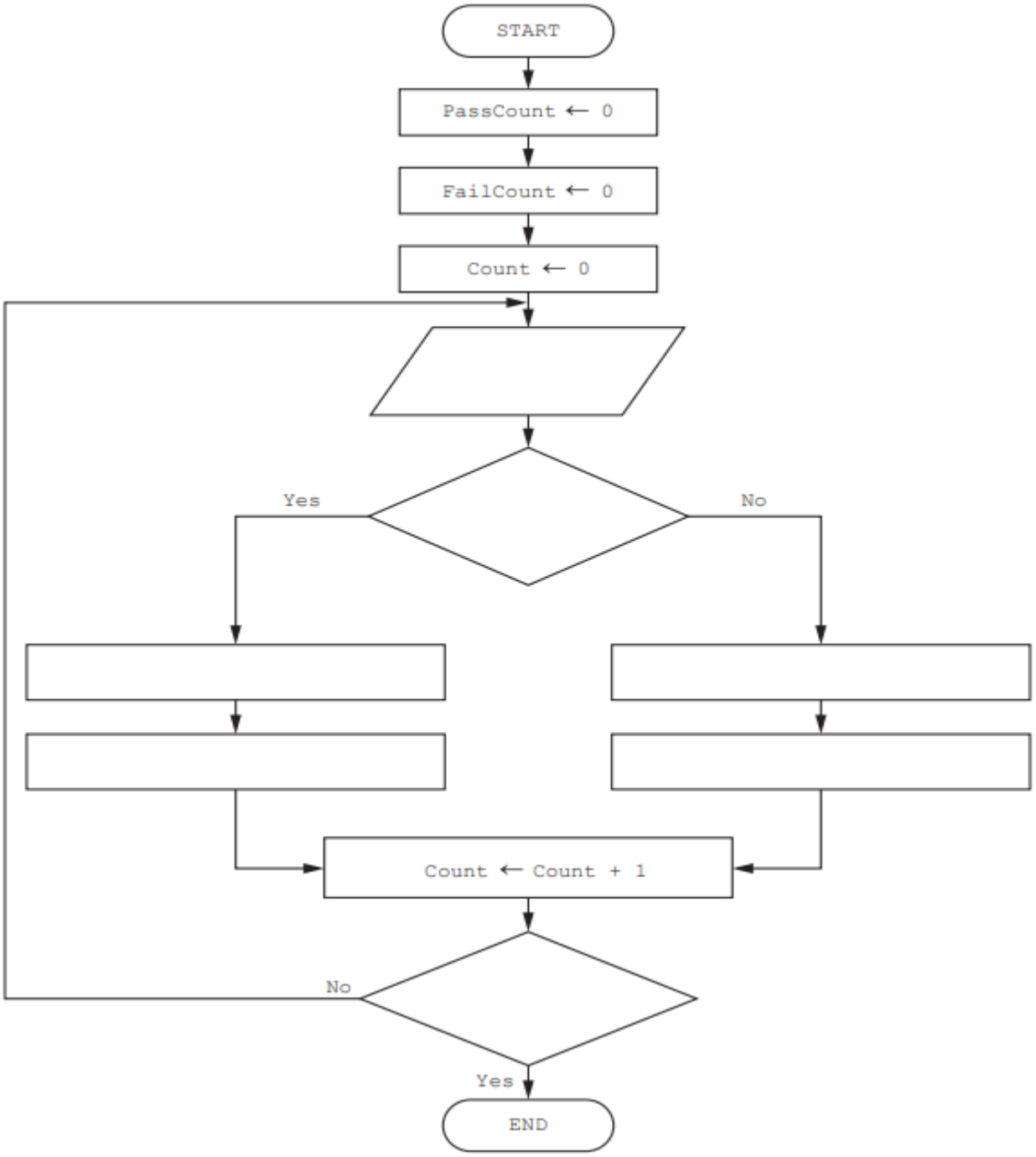
Complete a trace table for the algorithm using this input data:
88, 24, 60, 30, 44, 17, 25, 22, 54, 6, 999, −1

Counter	Pass	Mark	Help	OUTPUT

[5]

173 The flowchart shows an algorithm that should allow 60 test results to be entered into the variable `Score`. Each test result is checked to see if it is 50 or more. If it is, the test result is assigned to the `Pass` array. Otherwise, it is assigned to the `Fail` array.

(a) Complete this flowchart:



[6]

(b) Write a pseudocode routine that will check that each test result entered into the algorithm is between 0 and 100 inclusive.

.....

.....

.....

.....

.....

.....

.....

.....

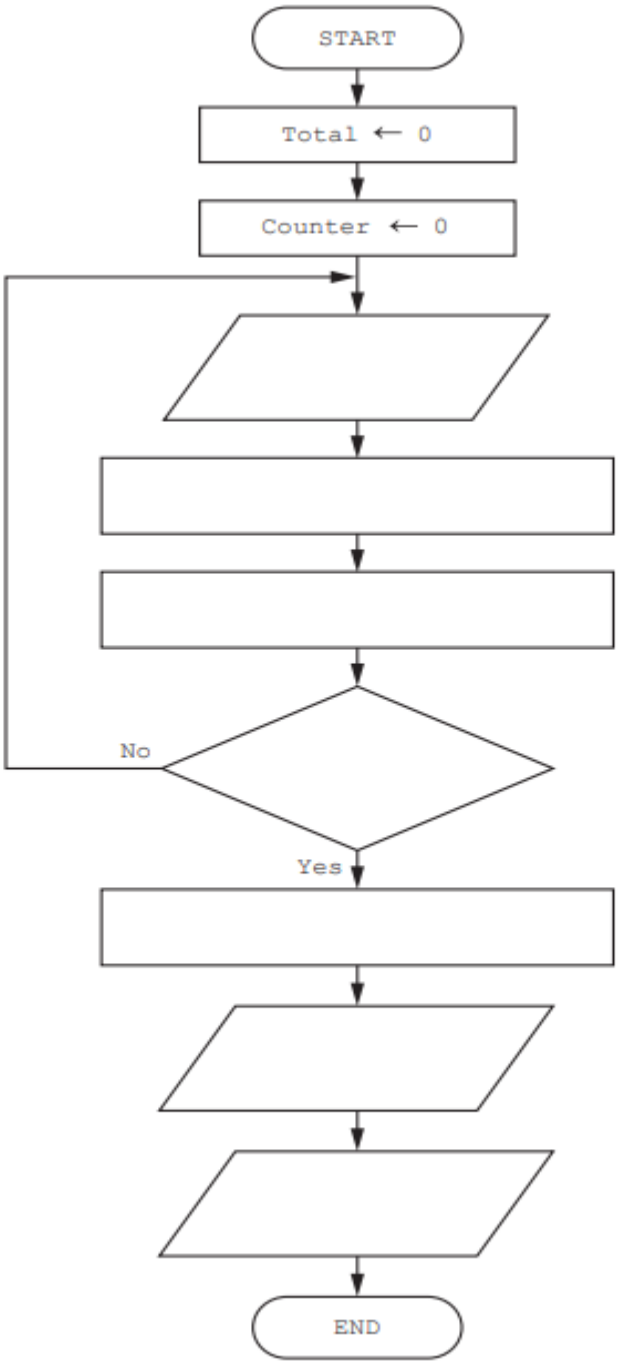
.....

..... [4]

174 The flowchart shows an algorithm that should:

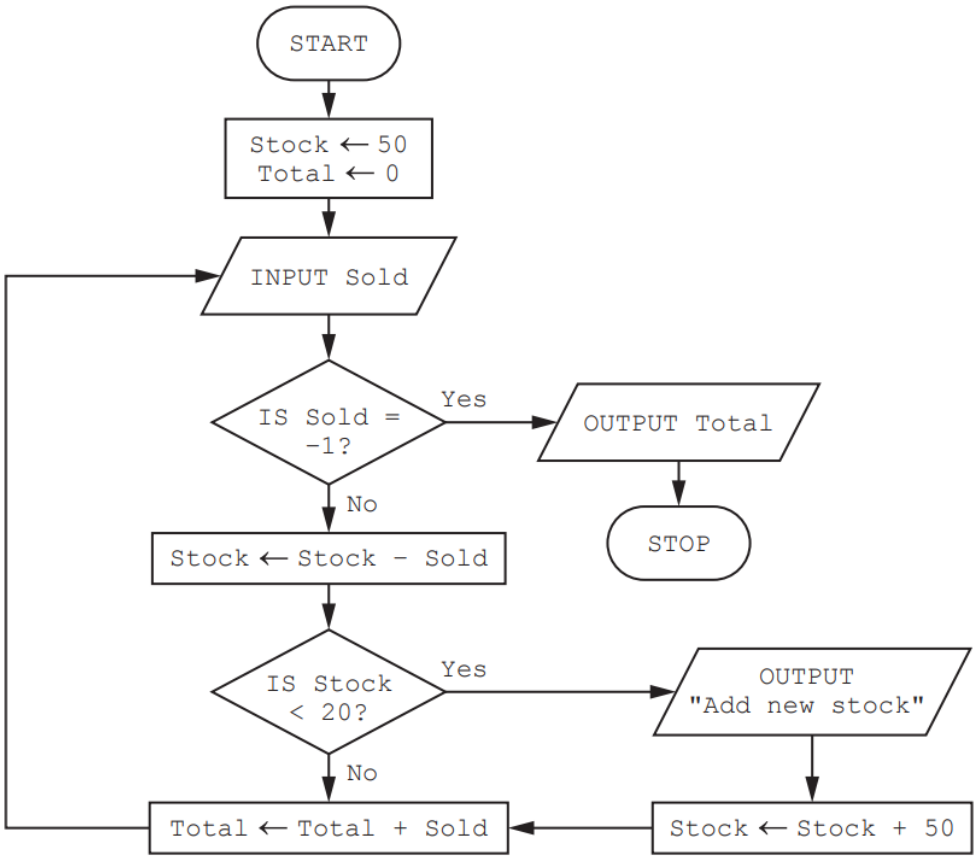
- allow 100 numbers to be entered into the variable `Number`
- total the numbers as they are entered
- output the total and average of the numbers after they have all been entered.

Complete this flowchart:



[6]

175 This algorithm makes sure that there are enough fresh bread rolls available for customers to buy.



(a) Complete the trace table for the algorithm using this input data:
24, 12, 6, 30, 12, 18, –1, 24

Sold	Stock	Total	OUTPUT

[4]

(b) Identify the problem that will occur if the input data starts with a value of 70.
Explain how you would correct this problem.

.....

.....

.....

.....

.....

.....

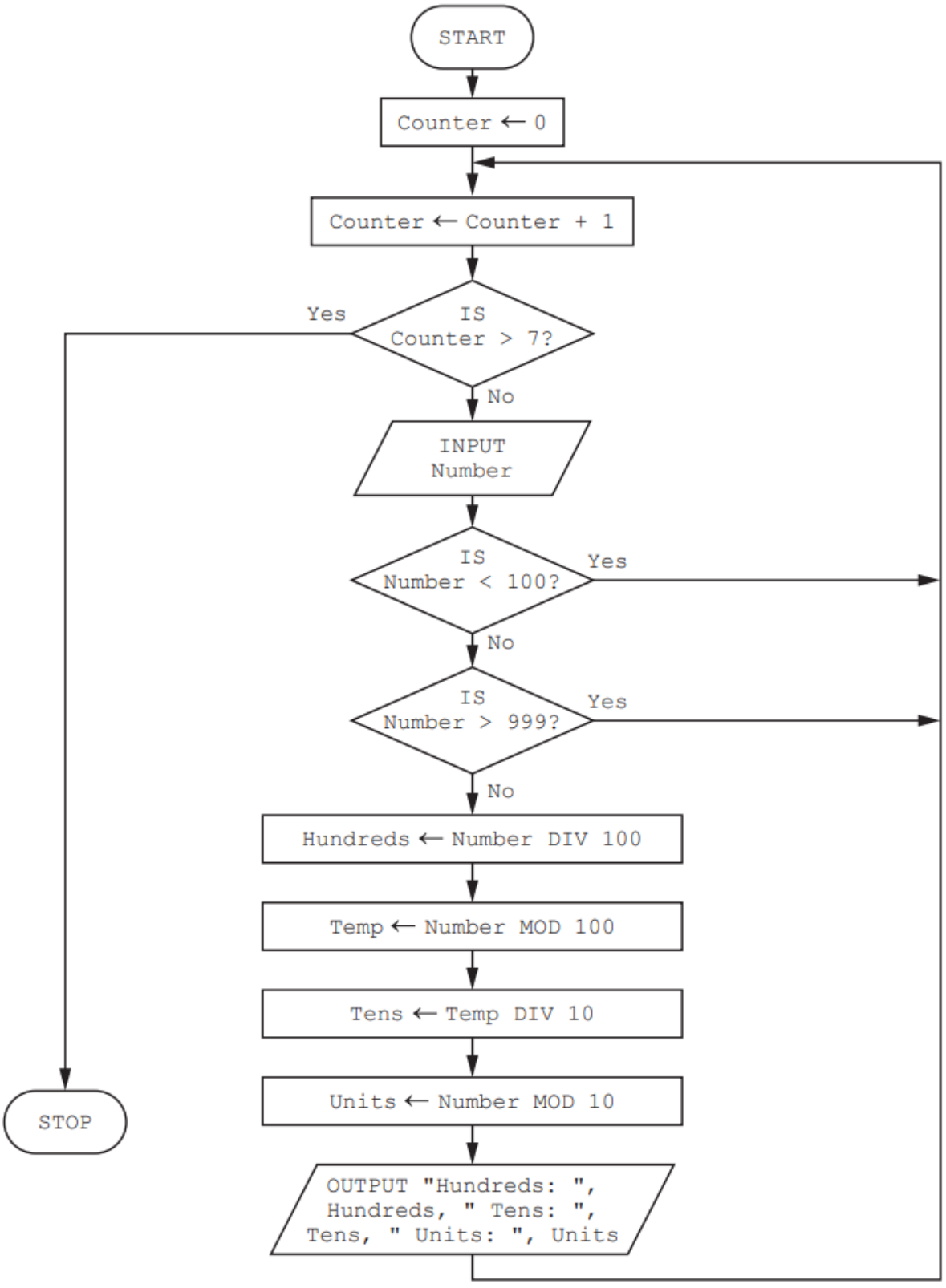
.....

..... [3]

176 This flowchart represents an algorithm to divide three-digit numbers into hundreds, tens and units.

The pre-defined function `DIV` gives the value of the result of integer division, for example `Y = 9 DIV 4` gives the value `Y = 2`

The pre-defined function `MOD` gives the value of the remainder of integer division, for example `R = 9 MOD 4` gives the value `R = 1`


START
Counter ← 0
Counter ← Counter + 1
IS Counter > 7?
Yes
No
INPUT Number
IS Number < 100?
Yes
No
IS Number > 999?
Yes
No
Hundreds ← Number DIV 100
Temp ← Number MOD 100
Tens ← Temp DIV 10
Units ← Number MOD 10
OUTPUT "Hundreds: ", Hundreds, " Tens: ", Tens, " Units: ", Units
STOP

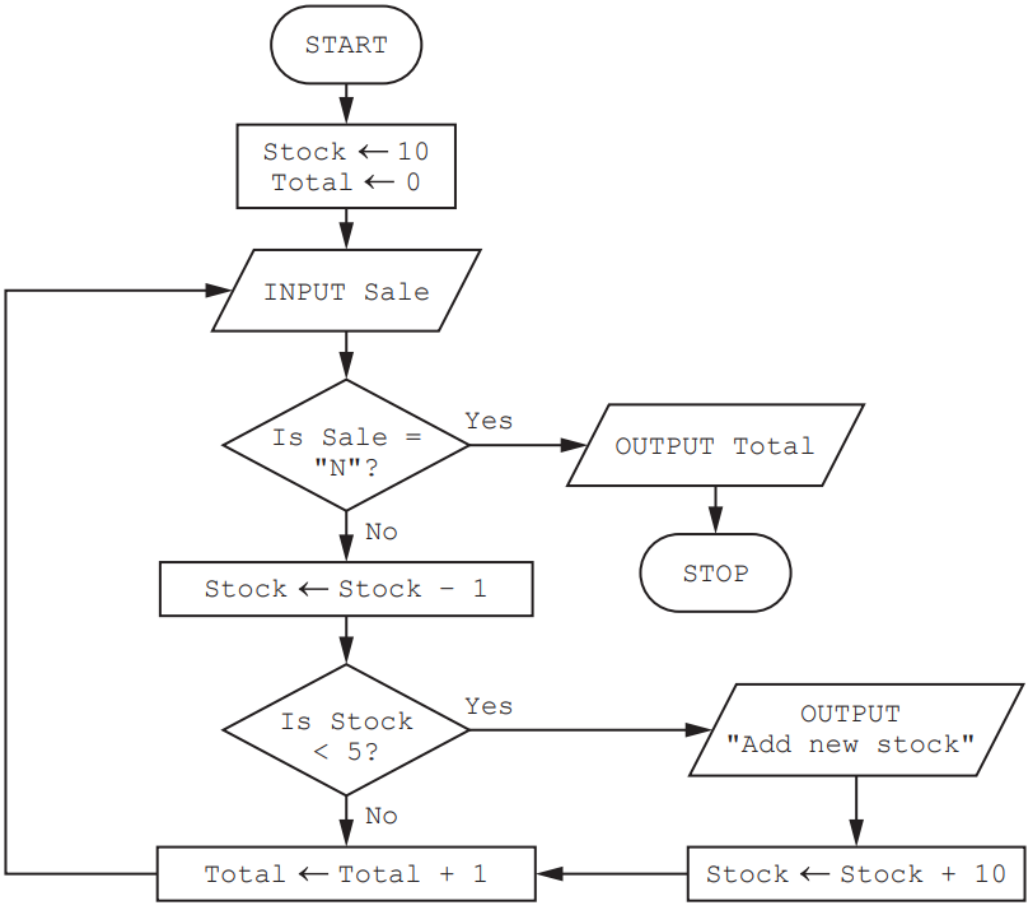
Complete the trace table for the algorithm using this input data:

97, 876, 4320, 606, 9875, 42, 124

Counter	Number	Hundreds	Temp	Tens	Units	OUTPUT

[5]

177 This algorithm makes sure that there are enough wheelbarrows in stock.



(a) Complete the trace table for the algorithm using this input data:

"Y", "Y", "Y", "Y", "Y", "Y", "N"

Stock	Total	Sale	OUTPUT

[4]

(b) Explain how you could extend the algorithm to allow for the sale of more than one wheelbarrow at a time.

.....

.....

.....

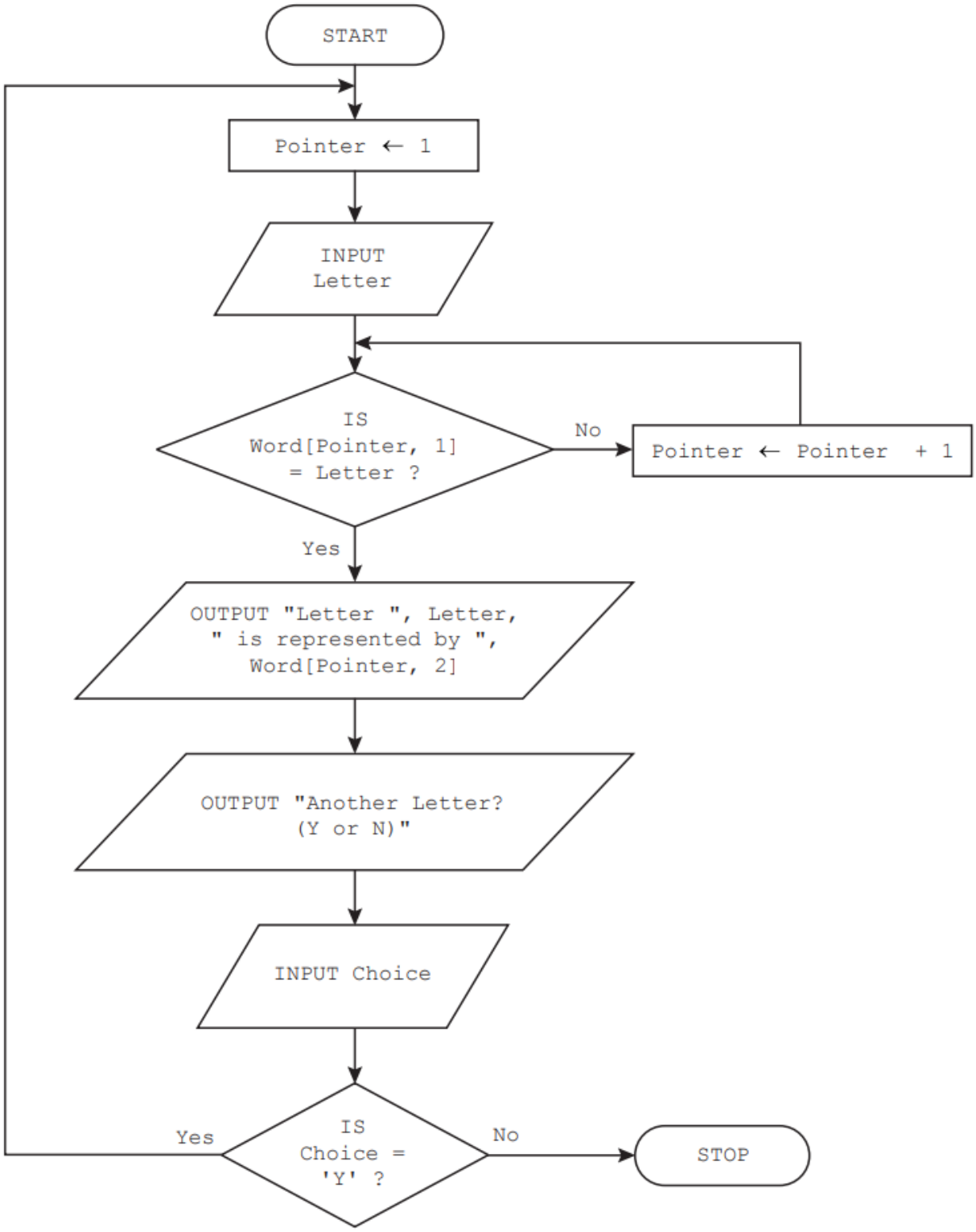
.....

.....

.....

..... [3]

178 The flowchart represents an algorithm.



The table represents the two-dimensional (2D) array `Word[]` which stores the first half of the phonetic alphabet used for radio transmission. For example, `Word[10,1]` is 'J'.

Index	1	2
1	A	Alpha
2	B	Bravo
3	C	Charlie
4	D	Delta
5	E	Echo
6	F	Foxtrot
7	G	Golf
8	H	Hotel
9	I	India
10	J	Juliet
11	K	Kilo
12	L	Lima
13	M	Mike

(a) Complete the trace table for the algorithm by using the input data: F, Y, D, N

Pointer	Letter	Choice	OUTPUT

[4]

(b) Identify the type of algorithm used.

.....

..... [1]

(c) Describe **one** problem that could occur with this algorithm if an invalid character was input.

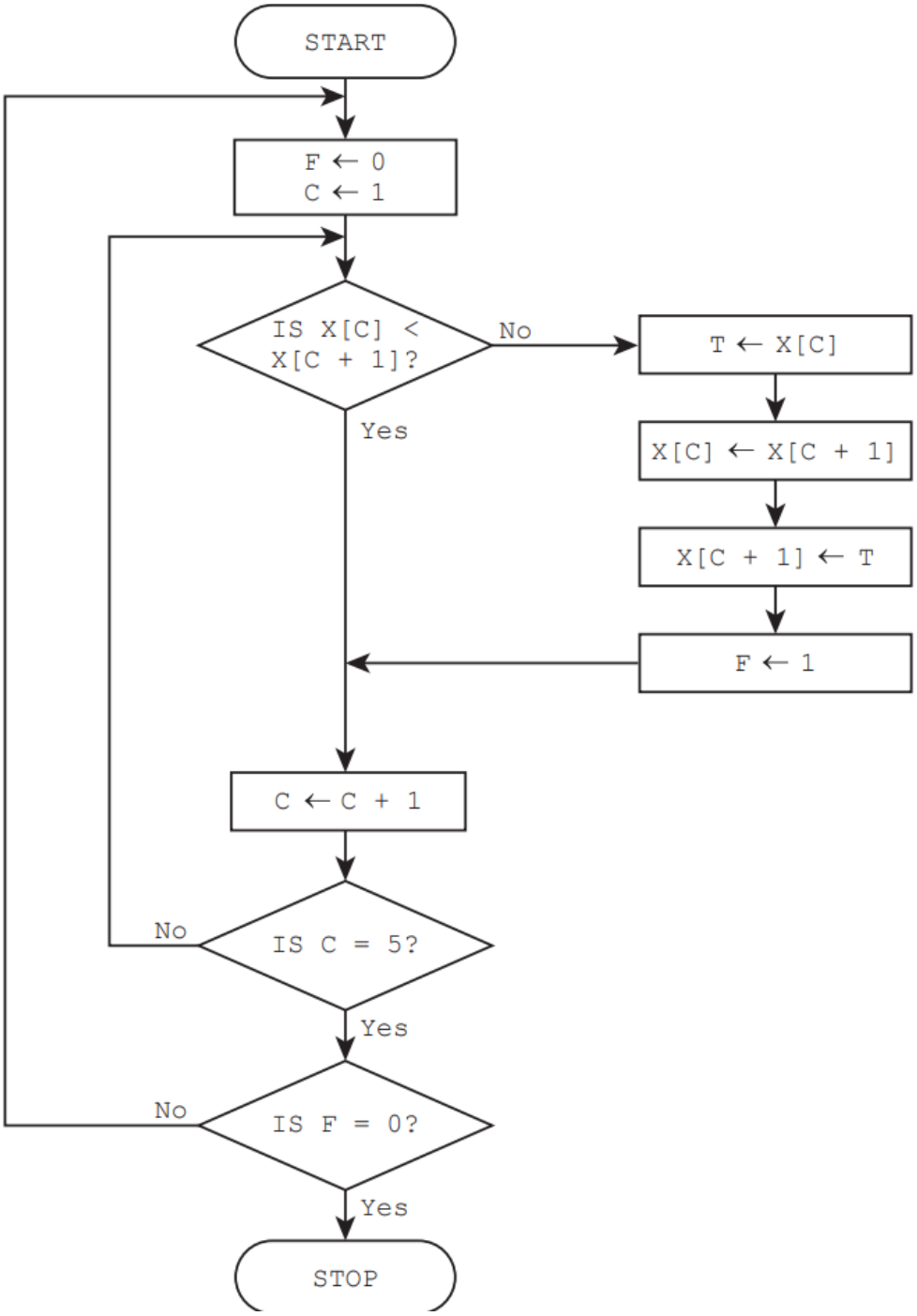
.....

.....

.....

..... [2]

179 This flowchart represents an algorithm.



(a) The array `X[1:5]` used in the flowchart contains this data:

X[1]	X[2]	X[3]	X[4]	X[5]
10	1	5	7	11

Complete the trace table by using the data given in the array.

F	C	X[1]	X[2]	X[3]	X[4]	X[5]	T
		10	1	5	7	11	

[5]

(b) Describe what the algorithm represented by the flowchart is doing.

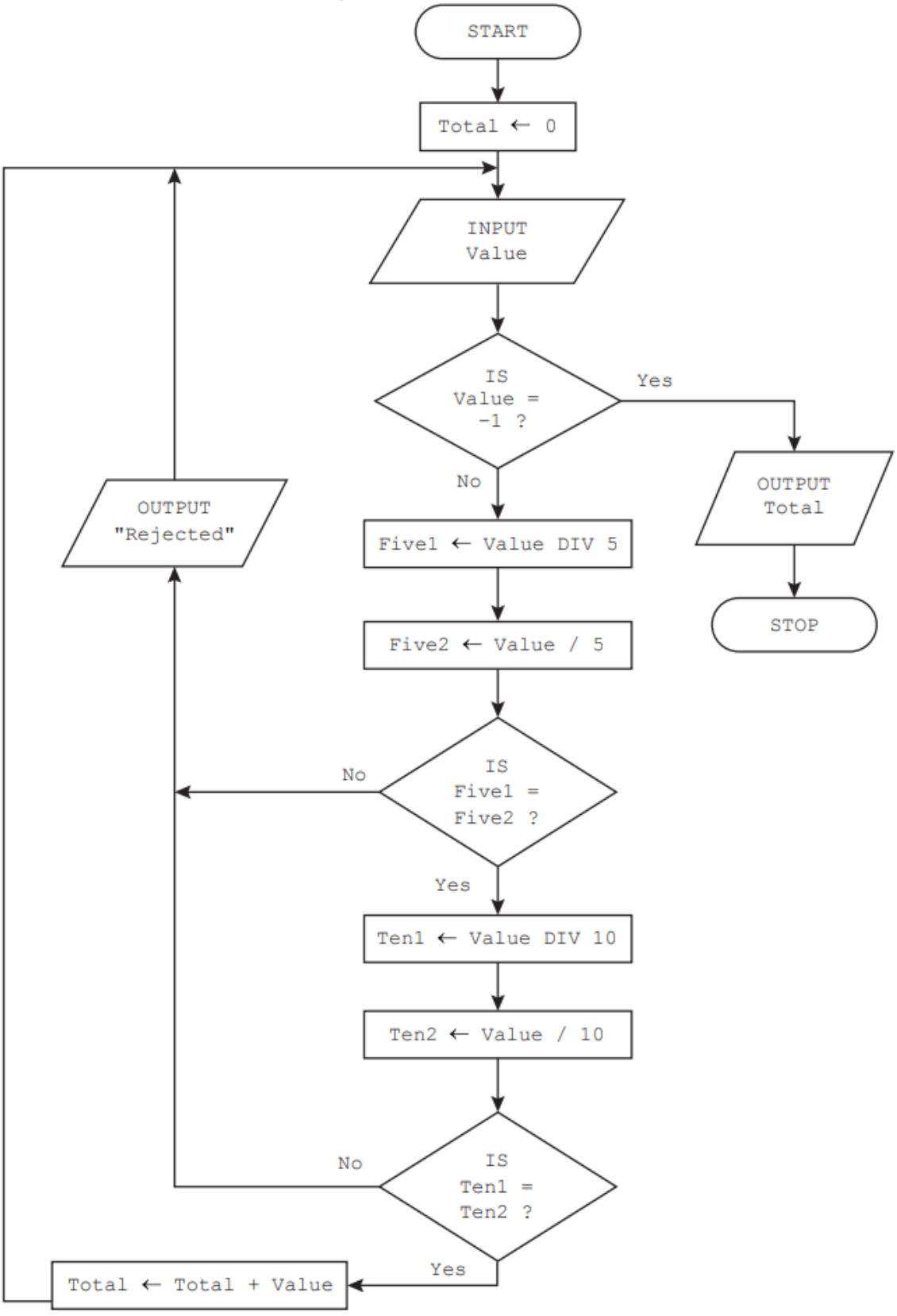
.....

.....

.....

..... [2]

180 The flowchart represents an algorithm.
An input of –1 will terminate the algorithm.



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

5, 50, 52, 555, 57, 500, –1, 5500, 55

Total	Value	Five1	Five2	Ten1	Ten2	OUTPUT

[6]

(b) Describe the purpose of the algorithm.

.....

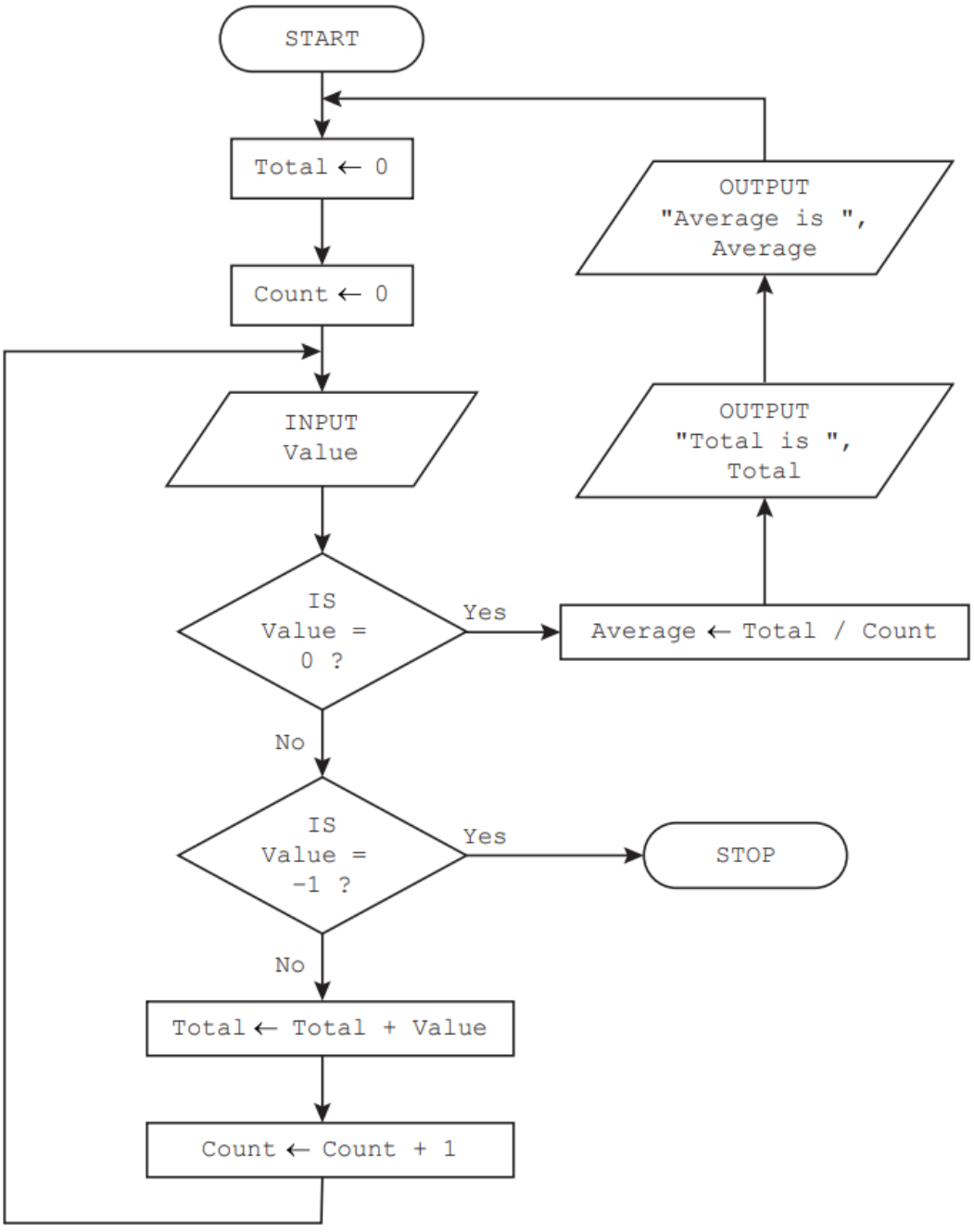
.....

.....

..... [2]

181 The flowchart represents an algorithm that performs a process on groups of values that are input. The algorithm will fail if the first value of any group is 0.

An input of –1 will terminate the algorithm.



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

25, 35, 3, 0, 57, 20, 25, 18, 0, –1, 307, 40, 0

Value	Average	Total	Count	OUTPUT

[5]

(b) Describe the purpose of the algorithm.

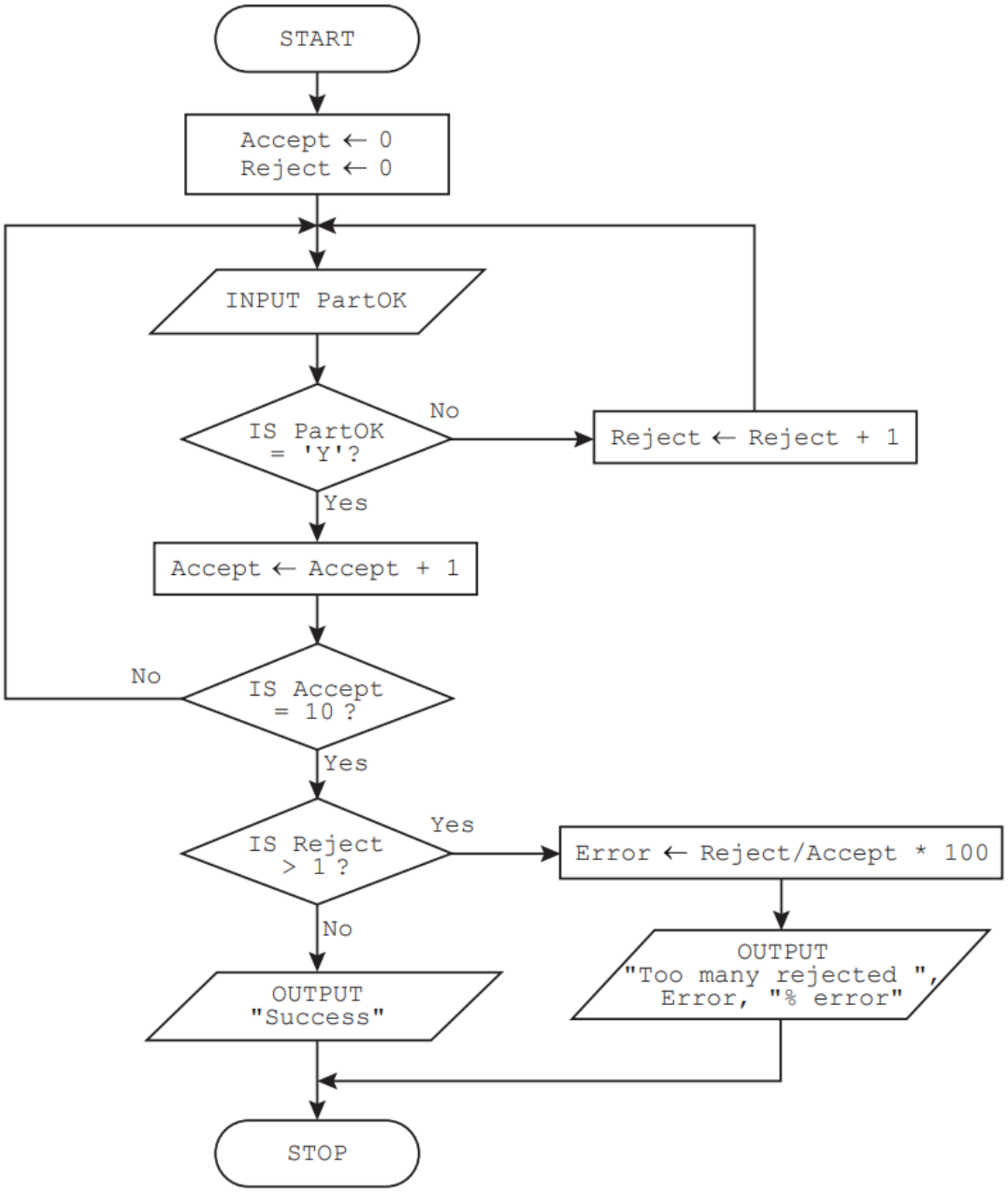
.....

.....

.....

..... [2]

182 This is an algorithm to find if a batch of parts has been manufactured successfully.



(a) Complete the trace table using this data:
Y, Y, Y, N, Y, Y, Y, Y, N, Y, Y, Y, Y

Accept	Reject	PartOK	Error	OUTPUT

[5]

(b) Describe how the algorithm should be changed to accept 'Y' or 'y' for a successfully manufactured part.

.....

.....

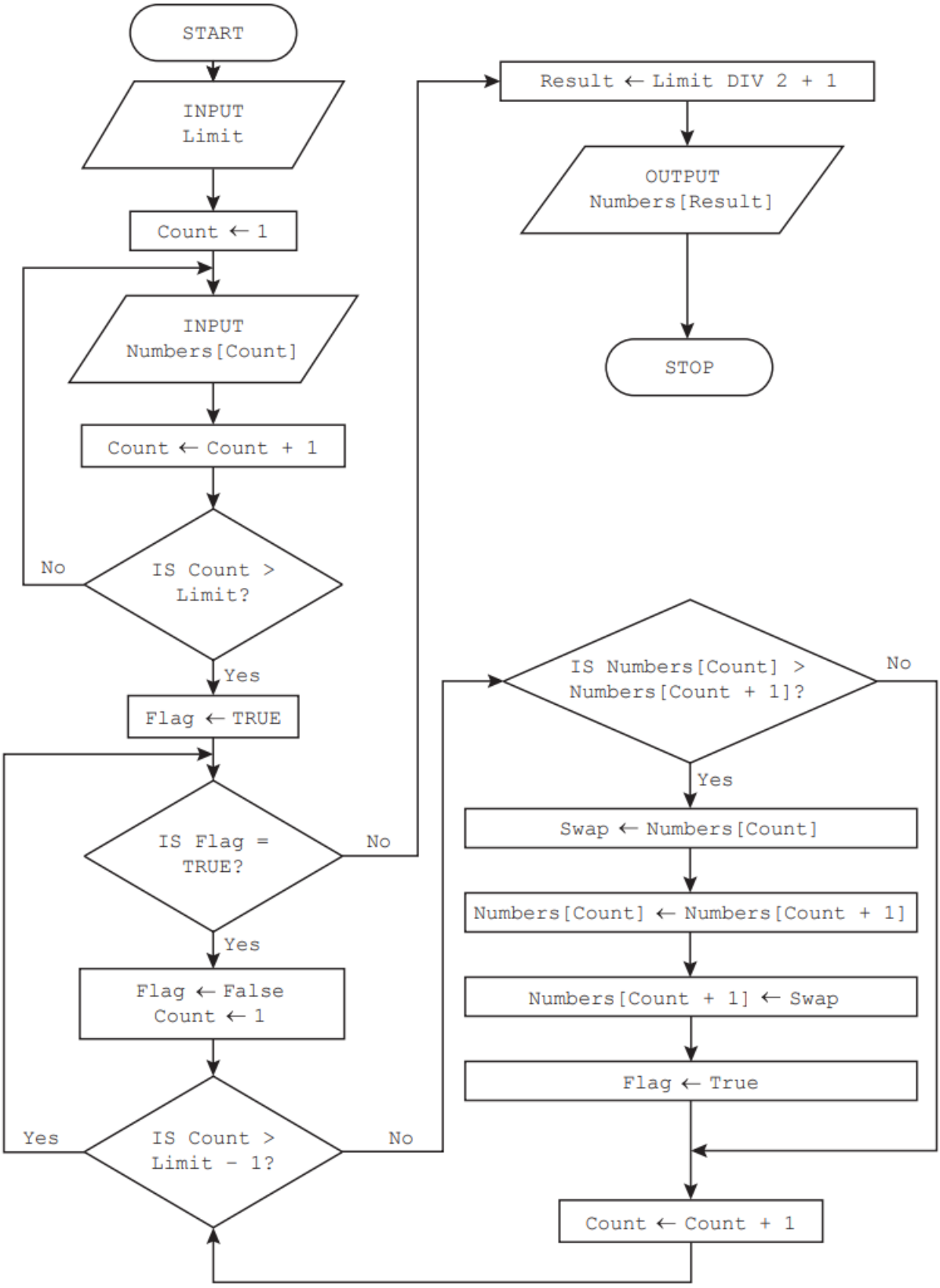
.....

.....

.....

..... [3]

183 The flowchart represents an algorithm.



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

7, 47, 50, 52, 60, 80, 63, 70

		Numbers										
Limit	Count	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Flag	Swap	Result	OUTPUT

[7]

(b) Outline the processes involved in the algorithm shown in the flowchart on **page 6**.

.....

.....

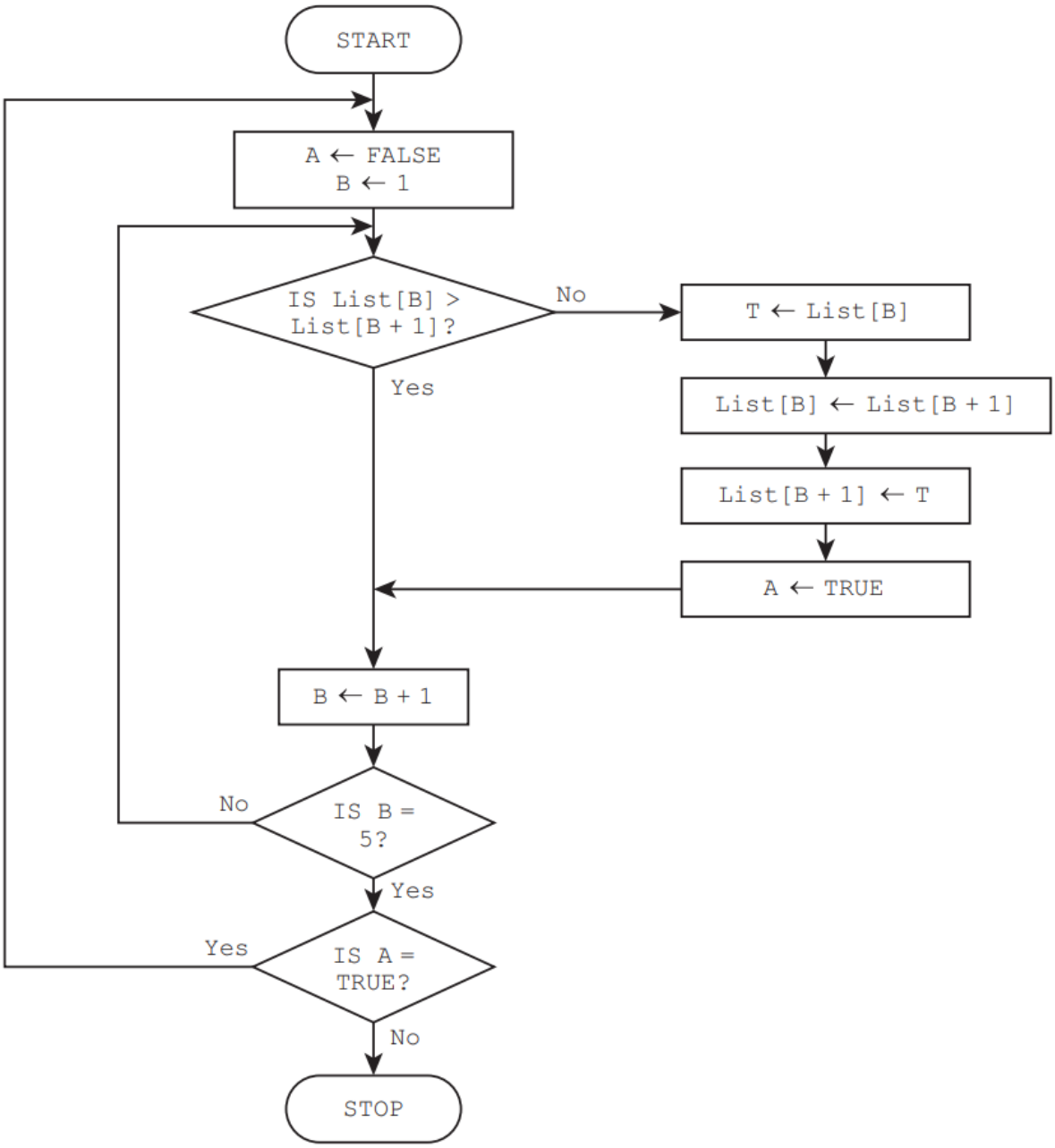
.....

.....

.....

..... [3]

184 This flowchart represents an algorithm.



(a) The array `List[1:5]` used in the flowchart contains this data:

List[1]	List[2]	List[3]	List[4]	List[5]
15	17	20	5	9

Complete the trace table using the data given in the array.

A	B	List[1]	List[2]	List[3]	List[4]	List[5]	T
		15	17	20	5	9	

[5]

(b) Describe what the algorithm represented by the flowchart is doing.

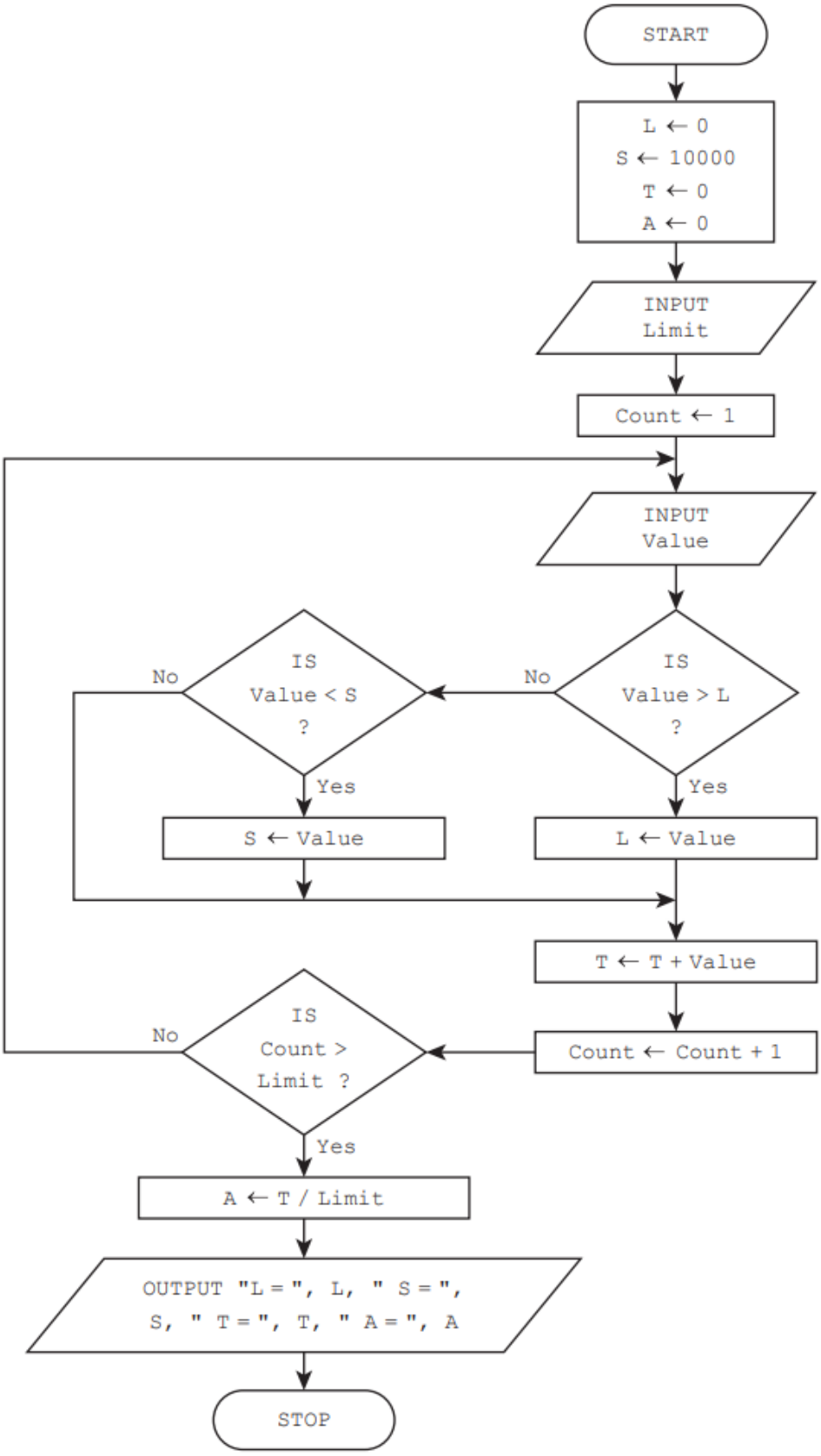
.....

.....

.....

..... [2]

185 The flowchart represents an algorithm.



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

10, 30, 18, 8, 25, 12, 17, 2, 50, 15, 5

L	S	T	A	Limit	Count	Value	OUTPUT

[6]

(b) Outline the purpose of the algorithm.

.....

.....

.....

..... [2]

(c) Explain why the identifiers `L`, `S`, `T` and `A` may **not** be appropriate to use as identifiers and how they could be improved.

.....

.....

.....

.....

.....

..... [3]

(d) State a more appropriate identifier for each of the variables `L`, `S`, `T` and `A`

Original identifier	Improved identifier
L	
S	
T	
A	

[2]

186 An algorithm has been written in pseudocode to check if a temperature is in a given range. The temperature values used in the algorithm are correct.

```
01 REPEAT
02     OUTPUT "Please enter temperature "
03     INPUT Temp
04     IF Temperature = 999
05       THEN
06         IF Temperature > 38.0
07           THEN
08             OUTPUT "Temperature too high"
09         ENDIF
10         IF Temperature < 35.0
11           THEN
12             OUTPUT "Temperature too low"
13         ENDIF
14         IF Temperature >= 35.0 OR Temperature <= 38.0
15           THEN
16             OUTPUT "Temperature normal"
17         ENDIF
18     ENDIF
19 WHILE Temperature = 999
```

(a) Identify the line numbers of **four** errors in the pseudocode and 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]

(b) Identify the temperature range used.

.....

.....

..... [2]

(c) Complete the trace table for the **corrected** algorithm using this data:

34.22, 36.1, 37.4, 38.0, 999, –1

Temperature	OUTPUT

[2]

187 This pseudocode represents an algorithm.

An input of –1 will terminate the algorithm.

```
DECLARE Count : INTEGER
DECLARE Answer : INTEGER
DECLARE Value : INTEGER
REPEAT
    INPUT Value
    IF Value <> -1
      THEN
        Answer ← Value
        FOR Count ← Value - 1 TO 1 STEP -1
            Answer ← Answer * Count
        NEXT Count
        OUTPUT Answer
    ENDIF
UNTIL Value = -1
```

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

5, 6, –1, 20, 9, 4

Value	Count	Answer	OUTPUT

[5]

(b) State the purpose of this algorithm.

.....

..... [1]

(c) Describe the problem that would be caused in this algorithm if a `Value` of 1, 0 or less than –1 was input.

.....

.....

.....

..... [2]

188 An algorithm has been written in pseudocode to check that a password meets a set of rules.

```
01 OUTPUT "Please enter password "
02 INPUT Password
03 Accept ← TRUE
04 IF LENGTH(Password) < 8 OR LENGTH(Password) > 20
05   THEN
06     Accept ← FALSE
07 ENDIF
08 IF LCASE(Password) = Password OR UCASE(Password) = Password
09   THEN
10     Accept ← FALSE
11 ENDIF
12 Index ← 1
13 Found ← FALSE
14 WHILE NOT Found AND Accept AND Index < LENGTH(Password)
15     IF SUBSTRING(Password, Index, 1) = '!'
16       THEN
17         Found ← TRUE
18     ENDIF
19     Index ← Index + 1
20 ENDWHILE
21 IF NOT Found
22   THEN
23     Accept ← FALSE
24 ENDIF
25 IF Accept
26   THEN
27     OUTPUT "Accepted"
28   ELSE
29     OUTPUT "Rejected"
30 ENDIF
```

(a) Complete the **three** trace tables using the data shown for each one.

Data: MYWORD

Password	Accept	Index	Found	OUTPUT

Data: M!word

Password	Accept	Index	Found	OUTPUT

Data: My!Hidden

Password	Accept	Index	Found	OUTPUT

[6]

(b) State the rules that the password must meet.

.....

.....

.....

.....

.....

..... [3]