Cambridge Assessment International Education

MEM

Cambridge IGCSE™

CANDIDATE NAME									
CENTRE NUMBER						CANDIDATE NUMBER				

5093781865

COMPUTER SCIENCE **0478/23**

Paper 2 Algorithms, Programming and Logic **May/June 2023**

1 hour 45 minutes

You must answer on the question paper.

No additional materials are needed.

INSTRUCTIONS

- Answer **all** questions.
- Use a black or dark blue pen. You may use an HB pencil for any diagrams or graphs.
- Write your name, centre number and candidate number in the boxes at the top of the page.
- Write your answer to each question in the space provided.
- Do **not** use an erasable pen or correction fluid.
- Do **not** write on any bar codes.
- Calculators must **not** be used in this paper.

INFORMATION

- The total mark for this paper is 75.
- The number of marks for each question or part question is shown in brackets [ ].
- No marks will be awarded for using brand names of software packages or hardware.

This document has **16** pages. Any blank pages are indicated.

[Turn over

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

A constant

A stores a value that can change at any time during the execution of a program. ☐

B stores a value that cannot change during the execution of a program. ☐

C stores values of multiple data types. ☐

D stores values that must be of the same data type. ☐

[1]

2 Explain the purpose of the library routines `MOD` and `RANDOM`

`MOD` .....

.....

.....

.....

`RANDOM` .....

.....

.....

.....

[4]

3 Describe what happens when a function is called during the execution of a program.

.....

.....

.....

.....

.....

..... [3]

4 **(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]

5 **(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]

6 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.

.....

.....

.....

.....

.....

.....

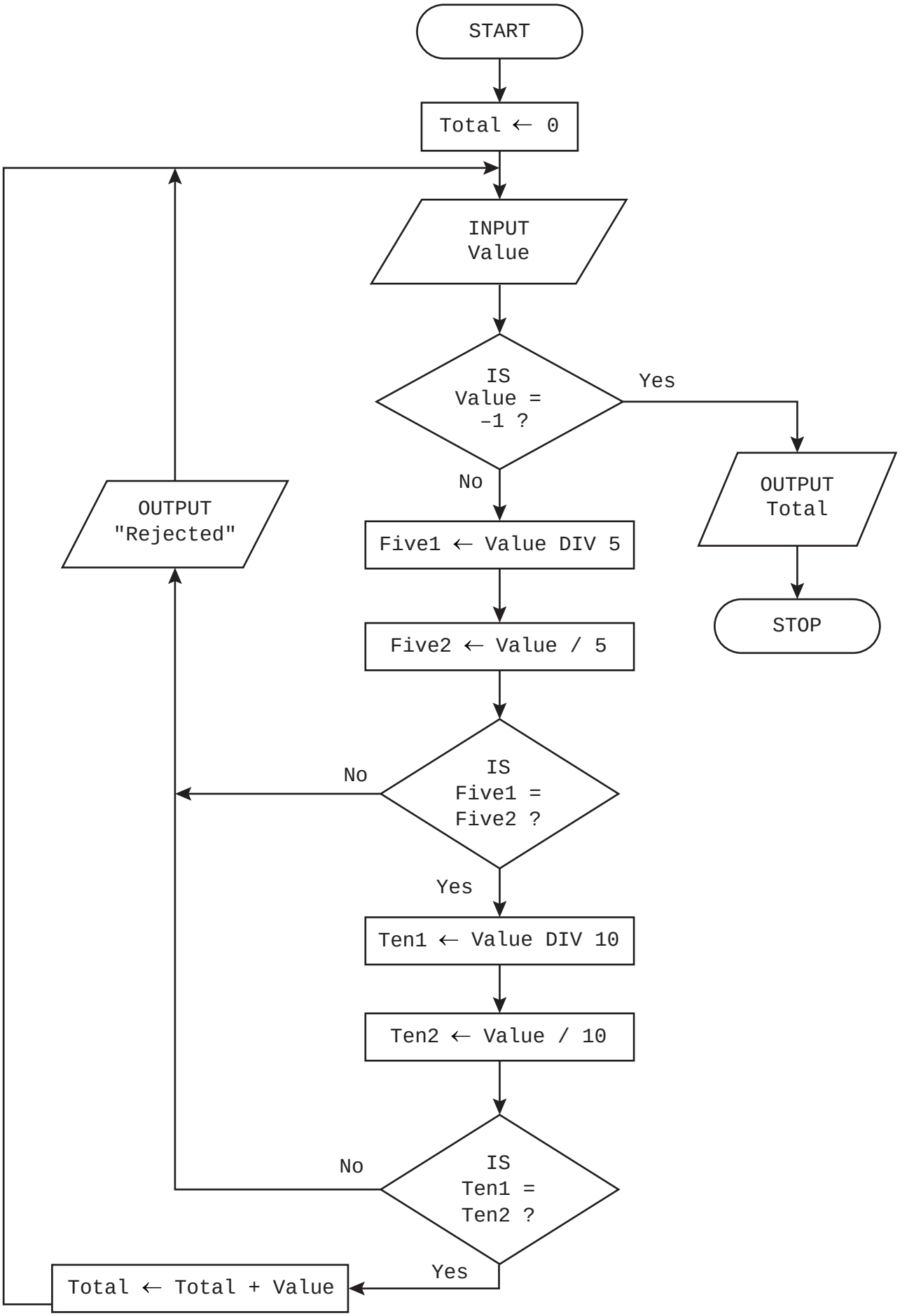
.....

.....

.....

..... [5]

7 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]

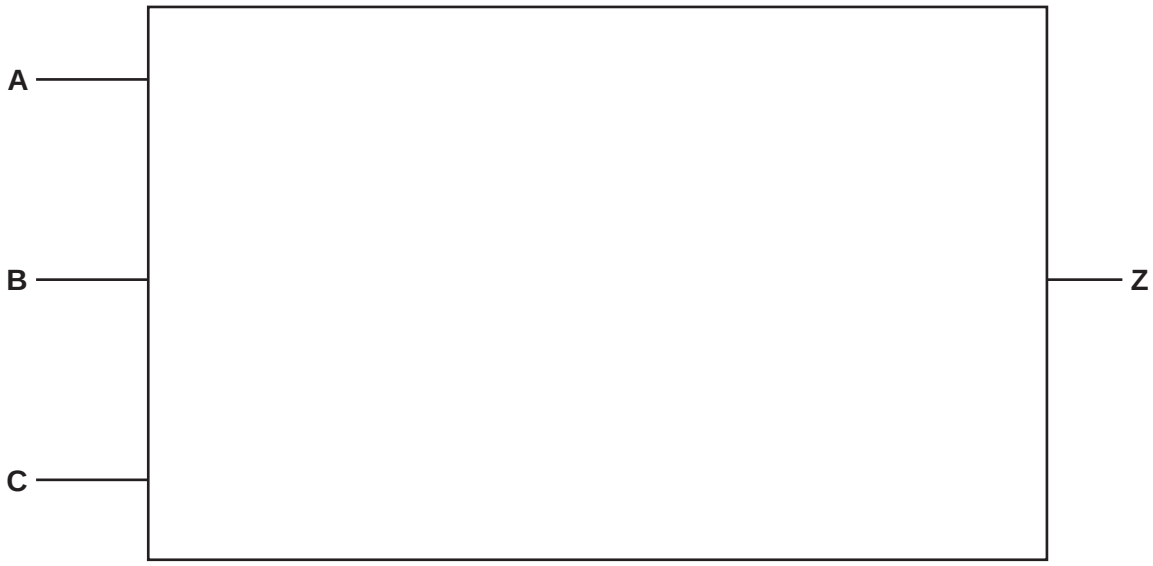
8 Consider the logic expression:

Z is 1 if (**A** = 1 AND **C** = NOT 1) AND (**B** = 1 NOR **C** = 1)

(a) Draw a logic circuit for this logic expression.

Each logic gate must have a maximum of **two** inputs.

Do **not** simplify this logic expression.



[4]

(b) Complete the truth table from the given logic expression.

A	B	C	Working space	Z
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

[4]

9 The variable `Saying` is used to store string data in a program.

(a) Write the pseudocode statement to declare the variable `Saying`

.....

..... [1]

(b) Write the pseudocode statements to:

- allow a string to be input to the variable `Saying`
- store the content of the variable `Saying` in a text file named `"Quotations.txt"`
- make sure the text file is closed at the end of the algorithm.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

10 A database table called `Site1` stores details of some holiday homes at a holiday park. The database shows the type of home, number of guests, whether it is privately owned and the weekly rate to hire it.

Name	Type	Private	Rate$	NumberGuest
Bay Lodge	Lodge	NO	1000	10
Bay View	Cabin	NO	400	4
Blue Skies	Cabin	NO	350	4
Cliff View	Cabin	NO	650	6
Coppice Lodge	Lodge	NO	1200	12
Green Lodge	Lodge	NO	1000	8
Henry	Cabin	YES	300	2
Hikers' Rest	Retreat	NO	750	6
Poppy	Cabin	NO	300	2
Summer Joy	Retreat	YES	750	6
Valley View	Cabin	NO	600	6
West Lodge	Lodge	YES	1200	12

(a) State the number of fields and the number of records in this database table.

Fields .....

Records .....
[2]

(b) Describe the purpose of a primary key.

.....

..... [1]

(c) The database uses the data types:

- alphanumeric
- character
- Boolean
- integer
- real
- date/time.

Complete the table to show the most appropriate data type for each field.

Field	Data type
`Type`	
`Private`	
`Rate$`	
`NumberGuest`	

[2]

(d) Give the output that would be produced by the structured query language (SQL) statement:

```
SELECT Name, NumberGuest, Rate$
FROM Site1
WHERE NumberGuest >= 10;
```

.....

.....

.....

.....

.....

..... [3]

11 A two-dimensional (2D) array `Contacts[]` is used to store names and telephone numbers. All the data is stored as strings. The array must have the capacity to store 100 contacts in the form of:

- column 1 – contact names as: last name, first name
  for example: Smith, John
- column 2 – telephone numbers.

The variable `CurrentSize` shows how many contacts are in the array.

Write a program that meets the following requirements:

- display a menu of choices:
  - enter new contact details
  - display all the contact details
  - delete all the contact details
- validate the menu input
- allow up to a maximum of five new contacts to be added to the array at any one time
- do **not** allow more than 100 contacts in total
- after new contacts have been added, sort the array by contact name, as long as there are at least two contacts in the array
- output the whole of the array
- delete the contents of the array.

You must use pseudocode or program code **and** add comments to explain how your code works.

You do **not** need to declare any arrays, variables or constants; you may assume that this has already been done.

All inputs and outputs must contain suitable messages.

You do **not** need to initialise the data in the array `Contacts[]` and the variable `CurrentSize`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [15]

BLANK PAGE

BLANK PAGE