



Cambridge IGCSE™

CANDIDATE
NAME

CENTRE
NUMBER

--	--	--	--	--

CANDIDATE
NUMBER

--	--	--	--



COMPUTER SCIENCE

0478/22

Paper 2 Algorithms, Programming and Logic

February/March 2024

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.

1 Tick (✓) **one** box to show which task is part of the analysis stage of the program life cycle.

- A coding
- B decomposition
- C design
- D testing

[1]

2 (a) **Four** test data types and **five** descriptions are shown.

Draw **one** line to link each test data type to its most appropriate description.

Test data type	Description
abnormal	a value that is accepted
boundary	a value that is the highest or lowest value to be accepted and the corresponding lowest or highest value to be rejected
	a value that is the highest or lowest value to be rejected
extreme	a value that is rejected
normal	a value that is the highest or lowest value to be accepted

[4]

(b) An algorithm has been written to test if an integer that is input is in the range 5 to 10 inclusive. Identify an example of suitable test data for each test data type.

- Abnormal
-
- Boundary
-
- Extreme
-
- Normal
-

[4]

4

(b) Write an algorithm in pseudocode to sort the array `Values[1:50]` into ascending order using a bubble sort.

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

..... [4]

4 Identify **three** data types used in programming.

1

2

3

[3]

- 5 An algorithm has been written in pseudocode to calculate the profit when an item is sold. Values for cost price and selling price are input, the profit is calculated (selling price – cost price) and output. The input of zero for either value stops the algorithm.

```

01 REPEAT
02     OUTPUT "Enter cost price "
03     INPUT Cost
04     OUTPUT "Enter selling price "
05     OUTPUT Sell
06     IF Cost <> 0 OR Sell <> 0
07         THEN
08             Profit ← Sell - Cost
09             OUTPUT "Profit is ", Profit
10     NEXT
11 UNTIL Cost = 0 OR Sell = 0

```

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

Error 1 line number

Correction

.....

Error 2 line number

Correction

.....

Error 3 line number

Correction

.....

[3]

(b) This algorithm needs to be improved with a range check.

(i) Write pseudocode to reject the input of values less than zero for variables `Cost` and `Sell`

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..... [3]

(ii) Describe **two** other validation checks that should be included in this algorithm.

Check 1

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

Check 2

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

[4]

6 Describe **two** types of iteration that a programmer can use whilst writing a program.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [4]

7 Consider the logic expression:

$$X = (A \text{ XOR } B) \text{ AND } (\text{NOT } B \text{ AND } C)$$

- (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.



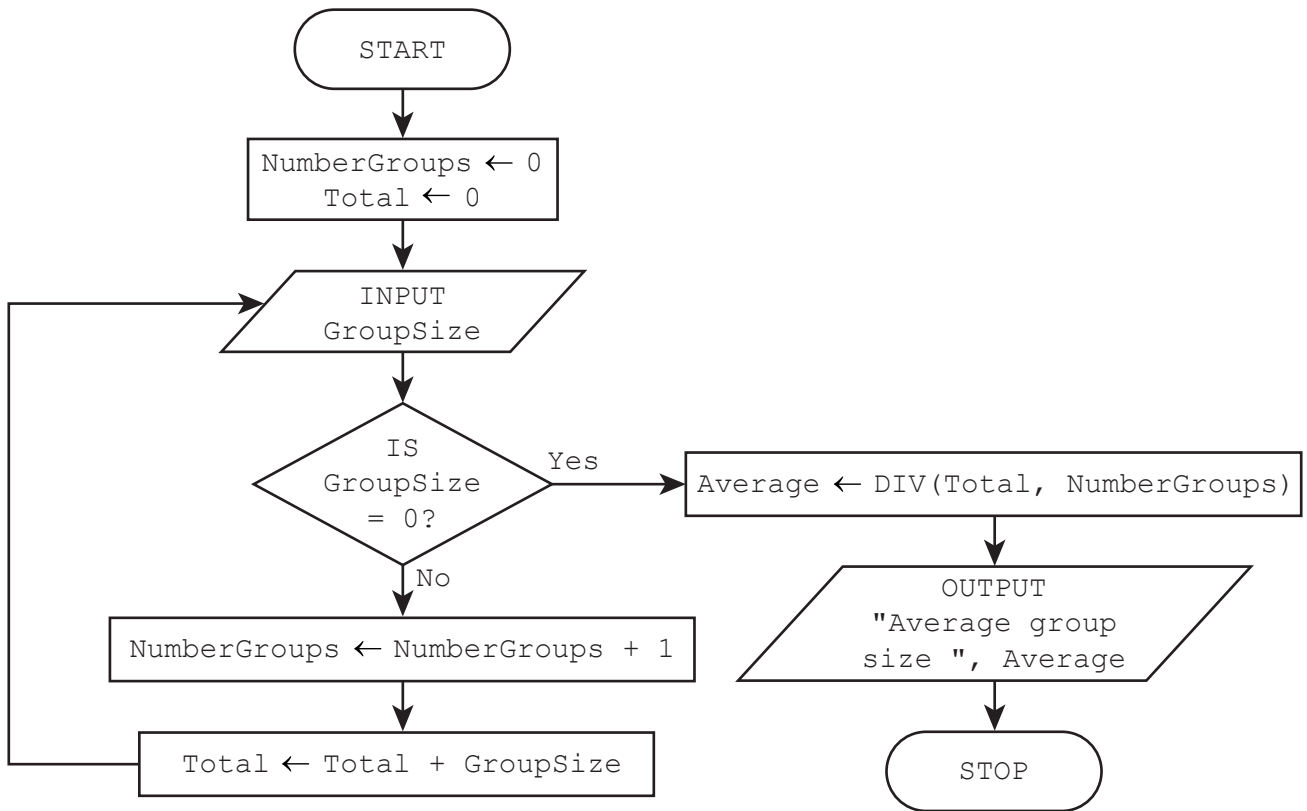
[4]

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

A	B	C	Working space	X
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]

8 This flowchart represents an algorithm to find the average size of groups of visitors to an attraction.



Complete the trace table using this data:

7, 10, 2, 8, 3, 9, 0, 6

NumberGroups	Total	GroupSize	Average	OUTPUT

[4]

9 A storage unit rental company wants to set up a new database table for the storage units that can be rented.

The table is called `StorageUnits` and needs to store these details:

- `SizeMetres` – size in square metres
- `Position` – first, second or third floor
- `Hoist` – whether there is a hoist available for the transfer of items
- `PriceMonth` – the price in dollars for a month’s rental
- `StorageID` – the code to identify each storage unit, for example S123

(a) (i) Give the name of the field that would be used for the primary key.

..... [1]

(ii) State the reason for choosing this field for the primary key.

.....
 [1]

(b) Complete the table to identify the most appropriate data type for these fields in the table `StorageUnits`

Field	Data type
<code>SizeMetres</code>	
<code>Position</code>	
<code>Hoist</code>	
<code>StorageID</code>	

[2]

(c) Complete this structured query language (SQL) statement to display only the storage code, price and size in square metres of all the storage units where there is a hoist available.

SELECT

FROM

WHERE ;

[4]

BLANK PAGE

- 11 Students in a class are recording the amount of time in minutes spent in front of a screen for each day of the week.

The one-dimensional (1D) array `StudentName[]` contains the names of the students in the class.

The two-dimensional (2D) array `ScreenTime[]` is used to input the number of minutes on each day spent in front of a screen.

The position of each student's data in the two arrays is the same. For example, the student stored at index 10 in `StudentName[]` and `ScreenTime[]` is the same.

The variable `ClassSize` contains the number of students in the class.

Write a program that meets these requirements:

- allows all the students to enter their daily minutes of screen times for the past week
- calculates the total number of minutes of screen time for each student in the week
- counts, for each student, the number of days with more than 300 minutes of screen time
- calculates the average weekly minutes of screen time for the whole class
- finds the student with the lowest weekly minutes of screen time
- outputs for each student:
 - name
 - total week's screen time in hours and minutes
 - number of days with more than 300 minutes of screen time
- outputs the average weekly minutes of screen time for the whole class
- outputs the name of the student with the lowest weekly screen time.

You must use pseudocode or program code **and** add comments to explain how your code works. All inputs and outputs must contain suitable messages.

Assume that the array `StudentName[]` and the variable `ClassSize` already contain the required data.

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

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

A series of horizontal dotted lines providing space for writing.

