* 0000800000001 *

Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME

CENTRE NUMBER | | | | | |

CANDIDATE NUMBER | | | | |

6786602078

COMPUTER SCIENCE **0478/23**

Paper 2 Algorithms, Programming and Logic **October/November 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.

DC (DE/SW) 336229/3
© UCLES 2024

[Turn over

1 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]

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

A abstraction ☐

B structure diagram ☐

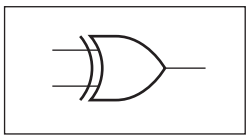
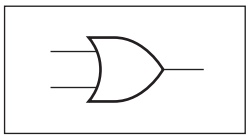
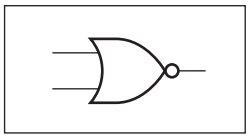
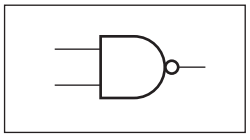
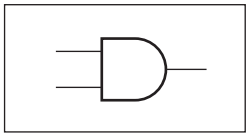
C test data ☐

D variable ☐

[1]

3 **Four** logic gates and **five** logic gate symbols are shown.

Draw **one** line to link each logic gate to its correct symbol. **Not** all logic gate symbols will be used.

Logic gate	Logic gate symbol
AND	
NAND	
NOR	
XOR	
	

[4]

DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN

4 Complete the paragraph about databases. Use terms from the list. You may need to use a term more than once. Some of the terms in the list will **not** be used.

fields	**comments**	**columns**	**constant**	**key**
table	**primary key**	**program**	**records**	**rows**
scripts	**database**	**validation**	**variable**	

Database tables consist of ..... and ..... .

Rows are ..... .

..... are fields.

Structured query language (SQL) ..... are used to query data.

A ..... uniquely identifies a record.

[6]

5 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]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

6 An incomplete algorithm has been written in pseudocode to count the number of values stored in an array and to find their average.
Values have been stored in the array starting at `A[1]`
All the values to be counted are non-zero.
A value of zero in the array indicates there are no more values stored.

```
01 DECLARE A : ARRAY[1:50] OF INTEGER

02 DECLARE C : INTEGER

03 DECLARE W : INTEGER

04 DECLARE X : INTEGER

05 W ← 0

06 C ← .....

07 WHILE A[C] <> 0

08     W ← .....

09     C ← C + 1

10 ENDWHILE

11 X ← .....
```

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

(b) Write pseudocode to display, with suitable messages:

- the number of values stored in the array
- the average of those values stored.

.....

.....

.....

.....

.....

..... [3]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

(c) Meaningful identifiers have **not** been used in this algorithm.

Suggest suitable meaningful identifiers for:

The array

A .....

The variables

C .....

X .....

W .....

[3]

7 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]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

8 A logic circuit is to be built to control the automatic opening of a window. The window **W** opens if the temperature **T** is too high, the heater **H** is off, and the air conditioning **A** is off.

air conditioning **A**	air conditioning off	0
	air conditioning on	1
temperature **T**	**not** too high	0
	too high	1
heater **H**	heater off	0
	heater on	1
window **W**	window closed	0
	window open	1

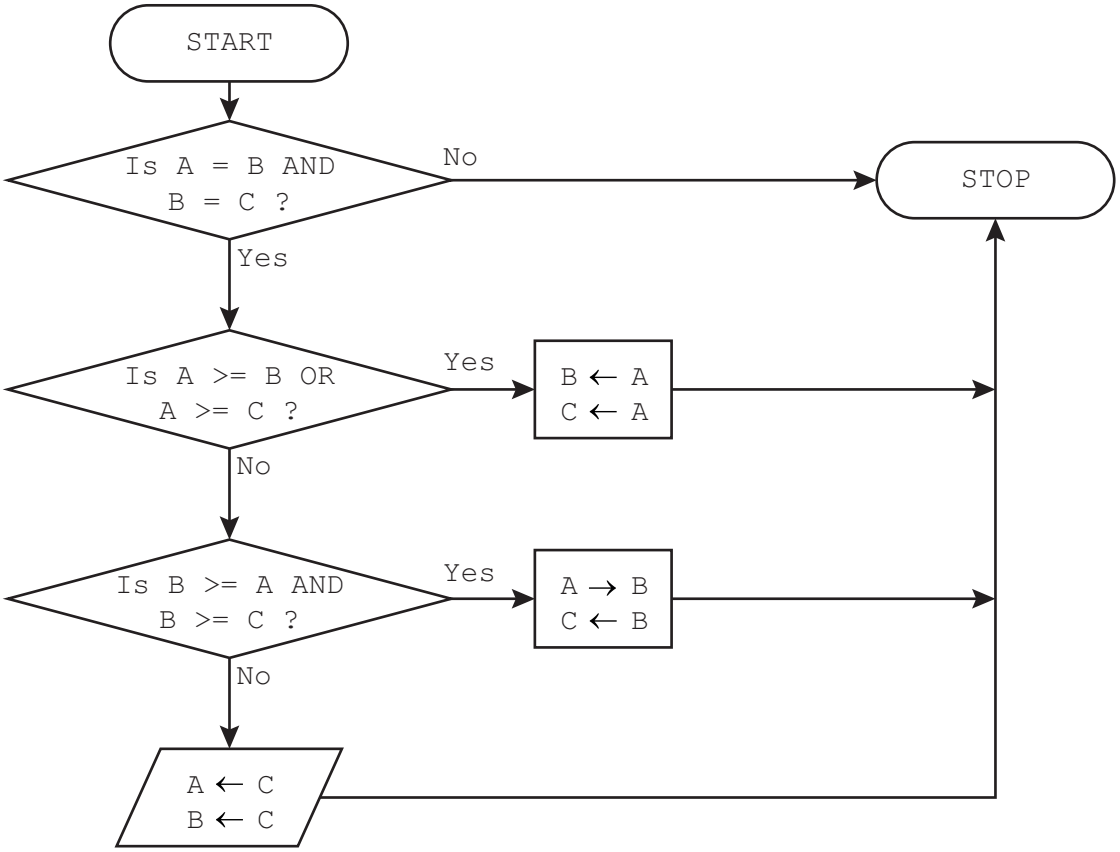
Complete the truth table for this problem.

A	**T**	**H**	**W**
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]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

9 A flowchart checks that values stored in three variables are identical.
If they are different, the highest value is stored in all three variables.



Identify **four** errors in the flowchart and suggest a correction for each error. You may label each error on the diagram with the corresponding error number.

Error 1 .....

Correction .....

.....

Error 2 .....

Correction .....

.....

Error 3 .....

Correction .....

.....

Error 4 .....

Correction .....

.....

[4]

DO NOT WRITE IN THIS MARGIN

10 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

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

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]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

11 Building materials are sold in bags. A new database table called `BuildStock` has been set up to store details about the materials for sale. Part of this table is given.

MtNo	Name	InStock	WeightKg	PricePerBag	NumberBags
MT01	Builders sand	Yes	50	4.50	50
MT02	Sharp sand	Yes	25	3.50	21
MT03	Red sand	No	50	2.75	0
MT04	Cement	No	25	6.85	0
MT05	Chippings	Yes	50	35.00	50
MT06	Cobbles	No	75	67.35	0
MT07	Pebbles small	Yes	50	34.50	3
MT08	Pebbles medium	Yes	25	25.50	10
MT12	Pebbles large	Yes	75	62.75	20
MT15	Washed gravel	Yes	50	12.75	12
MT16	Pea gravel	Yes	100	15.95	24

(a) Write the output from this structured query language (SQL) statement.

```
SELECT MtNo, Name
FROM BuildStock
WHERE WeightKg = 75
ORDER BY PricePerBag;
```

.....

.....

..... [3]

(b) (i) Complete this SQL statement to display only the names of all the materials that are out of stock.

```
SELECT .....
FROM .....
WHERE .....
```

[2]

(ii) Explain how another SQL statement using a different field could be used to display the same information.

.....

.....

.....

..... [2]

DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN DO NOT WRITE IN THIS MARGIN

BLANK PAGE

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

12 Members of a litter picking group complete a litter pick every month. Members' names are stored in one-dimensional (1D) array `PickerName[]`

Each member stores the weight of the litter they have picked in another one-dimensional (1D) array `PickedWeight[]`
The weights are in kilograms with one decimal place, for example 8.4

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

Every month, there is a small prize awarded to the members of the group who have the two heaviest weights. Certificates are awarded to all members with a pick weight of over three kilograms.

Write a program that meets the following requirements:

- allows the weight of members' picks to be input and validated
- sorts the arrays `PickedWeight[]` and `PickerName[]` in descending order of weight
- outputs the member names and the pick weights of the members with the two heaviest picks and identifies them as "Best in Group" and "Second best in Group"
- stores the names of all the members who will receive a certificate in the array `PickerCertificate[]`
- outputs a message stating the number of certificates to be printed.

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

You do **not** need to declare any arrays or variables; 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 `PickerName[]`

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS MARGIN

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [15]

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

BLANK PAGE

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

BLANK PAGE

DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN
DO NOT WRITE IN THIS MARGIN

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.