GradeMax

* 0000800000001 * DFD

Cambridge Assessment International Education

Cambridge IGCSE™

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

* 0 6 5 0 4 1 3 5 7 1 *

COMPUTER SCIENCE **0478/23**

Paper 2 Algorithms, Programming and Logic **October/November 2025**

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 (CJ) 342262/2
© UCLES 2025

[Turn over

GradeMax

1 Give **three** ways of creating a maintainable program.

1 .....

2 .....

3 .....

[3]

2 **Four** descriptions of programming techniques and **five** programming techniques are given.

Draw **one** line from each description to the relevant technique. **Not** all techniques will be used.

Description	Programming technique
adding up a set of numbers	output
displaying the result of a calculation	selection
executing a set of instructions based on a condition	totalling
executing the same set of instructions multiple times	iteration
	counting

[4]

3 Tick (✓) **one** box to identify the most appropriate data type to store `'M'`

A integer ☐

B char ☐

C real ☐

D Boolean ☐

[1]

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

GradeMax

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 **(a)** Tickets for a theme park are booked online. An adult ticket costs $12.99 and a child ticket costs $7.99. A booking fee of $1.99 is added to the total cost of each booking.

Write an algorithm in pseudocode to:

- input the number of adult tickets for a booking
- input the number of child tickets for a booking
- calculate and output the total cost of the booking using an appropriate message.

.....

.....

.....

.....

.....

.....

.....

..... [4]

(b) Each booking must have at least one adult ticket.

Explain how the algorithm could be changed to repeatedly ask for the number of adult tickets if the number input is less than 1.

Any code used must be fully explained.

.....

.....

.....

.....

.....

.....

.....

..... [4]

GradeMax

(c) The theme park has rules for a person to be able to go on a ride.

A person can go on a ride (**X** = 1) only if they:

- are 1.4 metres tall or over
- are 12 years old or over
- do **not** have a medical condition.

Input	Description	Binary value
A	under 1.4 metres	0
	1.4 metres or over	1
B	under 12 years old	0
	12 years old or over	1
C	does **not** have a medical condition	0
	has a medical condition	1

(i) Write the logic expression for this problem.

X = ... [2]

(ii) Complete the truth table for this problem.

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]

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

GradeMax

5 A program has been written in pseudocode to input five numbers between 1 and 900 inclusive and store the highest number and the lowest number entered.

```
01  Lowest ← 900
02  Highest ← 1
03  FOR Count ← 1 TO 5
04      INPUT Number
05      IF Number < Lowest
06        THEN
07          Lowest ← Number
08        ELSE
09          IF Number > Highest
10            THEN
11              Highest ← Number
12          ENDIF
13      ENDIF
14   NEXT Count
```

(a) Complete the trace table for the algorithm using the following numbers:

563, 21, 376, 99, 400

Count	Number	Lowest	Highest

[4]

(b) The algorithm does **not** work correctly.

State the reason why the algorithm does **not** work as expected and explain how the algorithm could be corrected.

Any code used must be fully explained.

Reason .....

.....

Correction .....

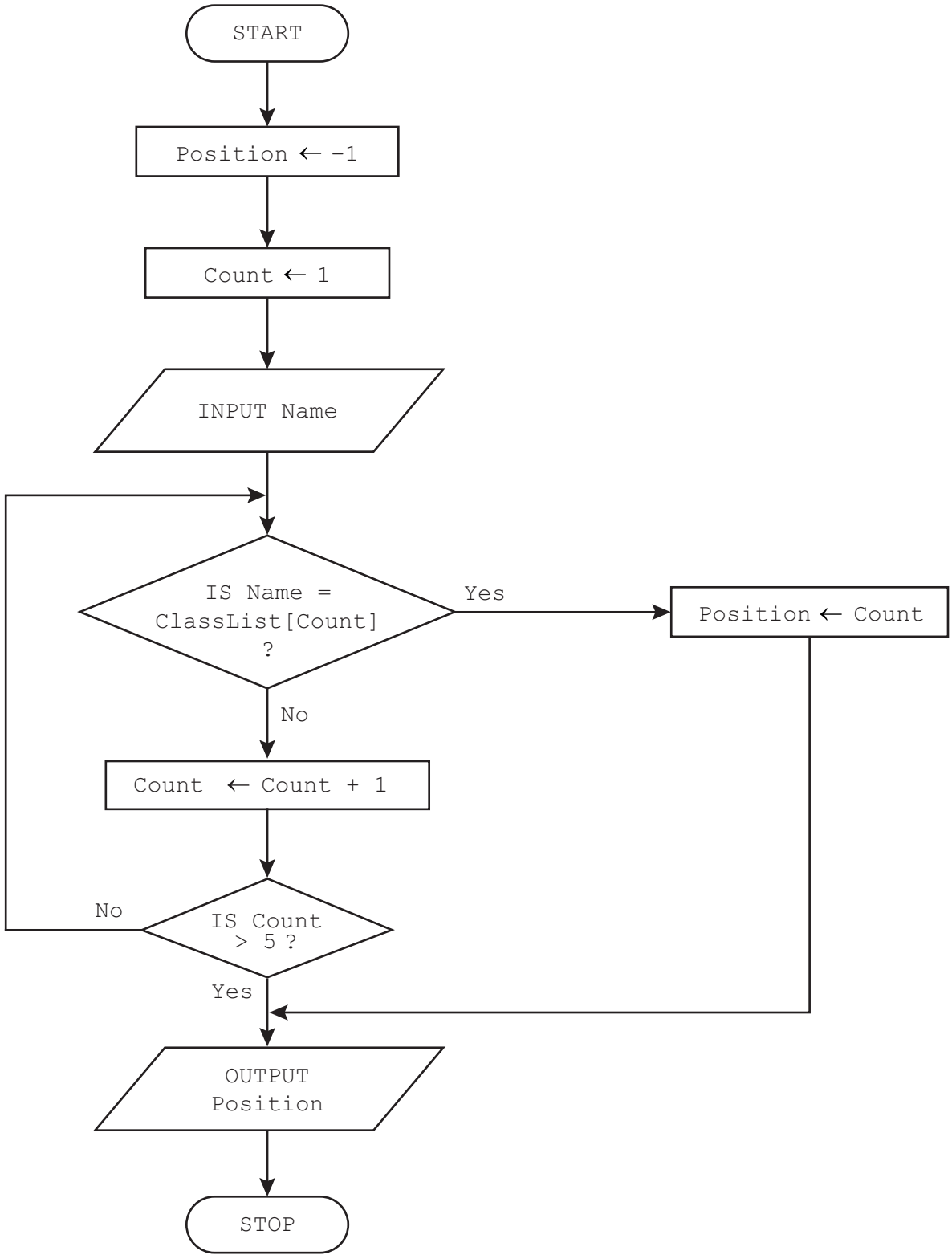
.....

[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

GradeMax

6 The flowchart represents an algorithm.



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

GradeMax

(a) The algorithm uses the one-dimensional (1D) array `ClassList` that contains the following data:

ClassList

1	2	3	4	5
Henil	Sven	Ella	Asia	Ridwan

Complete the table by writing the output from the flowchart when each of the given names are input.

Name	OUTPUT
Ridwan	
Yan	
Ella	

[2]

(b) Identify the standard method of solution used by the flowchart.

..... [1]

(c) Write the algorithm in pseudocode.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [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

GradeMax

7 An algorithm has been written in pseudocode to:

- input a whole number between 1 and 255 inclusive
- convert the input to an 8-bit binary number
- store each bit of the binary number in an array.

For example, if the denary number 127 was input, the binary number 01111111 would be stored in the array.

```
01  DECLARE BinaryArray : ARRAY[1:8] OF INTEGER
02  DECLARE DenaryNumber : CHAR
03  OUTPUT "Please enter a whole number between 1 and 255 inclusive "
04  INPUT DenaryNumber
05  FOR Count ← 7 TO 1 STEP -1
06      BinaryArray[Count] ← MOD(DenaryNumber,3)
07      DenaryNumber ← DIV(DenaryNumber,2)
08  NEXT Count
```

(a) 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]

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

GradeMax

(b) Identify **one** programming technique used in the algorithm.

..... [1]

(c) **(i)** The algorithm can accept a number between 1 and 255 inclusive as input.
Two validation checks that could be applied to `DenaryNumber` are a presence check and a type check.

Identify **one** other validation check that could be applied to `DenaryNumber`

..... [1]

(ii) Write the pseudocode to implement the validation check you identified in **(c)(i)**.

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

.....

.....

.....

.....

.....

..... [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

GradeMax

8 A school has set up a database table `EquipmentLoan` to record details of school equipment loaned to students.

EquipmentID	Description	Name	DateLoaned	Returned
Cam01	Digital camera	James	23/04/2025	No
Cam04	Digital camera	Amir	12/03/2025	Yes
Gra02	Graphics tablet	Nikita	27/04/2025	No
Tab01	Tablet	Holly	18/03/2025	Yes
Ext01	External Hard drive	Hussein	27/04/2025	Yes
Cam02	Digital camera	Emily	10/04/2025	Yes
Tri01	Tripod	Emily	10/04/2025	Yes

(a) (i) Give the name of the field that is most appropriate to be the primary key.

..... [1]

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

.....

..... [1]

(b) Complete the structured query language (SQL) statement to list equipment **not** returned. The list must contain only the descriptions and student names.

```
SELECT .....

FROM .....

WHERE ..... ;
```

[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

GradeMax

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 The variables `Name`, `Age` and `Found` are used in a computer program.
`Name` holds the name of a person. `Age` holds the age of a person as a whole number.
`Found` holds a flag that can be set to `TRUE` or `FALSE`

(a) Write pseudocode statements to declare the variables `Name`, `Age` and `Found`

.....

.....

..... [3]

(b) The procedure `StoreData(Name)` stores the name of a person in the file `People.txt`

(i) Write pseudocode statements to:

- define the procedure with the parameter
- store the name of the person in the file.

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... [5]

(ii) Write the pseudocode that uses the procedure `StoreData()` to store the name 'Suella' in the file `People.txt`

.....

.....

..... [2]

GradeMax

10 A number game has two players. Random whole numbers are generated for each player. The numbers are compared and players are given points based on the result of the comparison.

The two-dimensional (2D) array `NumberGenerated[]` stores each number generated for each player. The first dimension stores the player number and the second dimension stores the random whole numbers generated.

The game is played as follows:

- 100 random whole numbers between 1 and 6 inclusive are generated for player 1 and stored in the array.
- 100 random whole numbers between 1 and 6 inclusive are generated for player 2 and stored in the array.
- The first number generated for each player is compared and the player with the higher number is awarded 2 points. If the two numbers are the same, both players are awarded 1 point each.
- The comparisons repeat for all 100 numbers generated.
- The player with the higher total number of points is output as the winner.
- If the players have the same total number of points, a new random whole number is generated for each player and compared. This is repeated until one player has a higher number than the other player. 2 points are added to that player's total number of points, making this player the winner.

Write a program that meets the following requirements:

- input and validate the names of the two players
- generate 100 random whole numbers between 1 and 6 inclusive for each player and store them in the array `NumberGenerated[]`
- calculate and store the total number of points for each player
- output the name and total number of points for each player in descending order of total number of points.

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; assume that this has already been done.

All outputs must contain suitable messages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

© UCLES 2025

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

GradeMax

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

GradeMax

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

GradeMax

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]

GradeMax

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.