CAMBRIDGE
International Education

# Cambridge O Level

CANDIDATE NAME

CENTRE NUMBER

CANDIDATE NUMBER

*5519421014*

**COMPUTER SCIENCE** **2210/21**

Paper 2 Algorithms, Programming and Logic **May/June 2026**

**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. Do **not** use correction fluid or tape.
- 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 (LO) 369886
© Cambridge University Press & Assessment 2026

**1** Tick (✓) **one** box to identify the type of validation check to test whether a number entered into a computer system is between two set values.

**A** length check ☐

**B** presence check ☐

**C** type check ☐

**D** range check ☐

[1]

**2** Identify **four** operators that are **not** arithmetic operators from the list.

```
/          >          <=          +          *
^          AND        DIV         MOD        NOT
```

1 .....

2 .....

3 .....

4 .....

[4]

**3** Complete the table by writing the most appropriate data type for each variable described.

| variable | description | example data | data type |
|---|---|---|---|
| `Age` | the age of the student | `16` | |
| `TelephoneNumber` | the telephone number of the student | `+44 7846398573` | |
| `Teenager` | whether the student is a teenager or not | `TRUE` | |

[3]

DO NOT WRITE IN THIS MARGIN

**4** The following pseudocode algorithm uses a bubble sort to sort the array `Names[50]`

Complete the algorithm.

```
Last ← .....
REPEAT
    Swap ← .....
    FOR ..... ← 1 TO Last - 1
        IF Names[Index] > Names[Index + 1]
          THEN
            Temp ← Names[Index]
            Names[Index] ← Names[Index + 1]
            Names[Index + 1] ← Temp
            Swap ← TRUE
        ENDIF
    NEXT Index
    Last ← Last - 1
UNTIL ..... Swap OR Last = 1
```

[4]

DO NOT WRITE IN THIS MARGIN

**5** A computer system automatically creates usernames for new users. A new user enters their first name and their last name.

The system:

- takes the first letter of the first name
- takes the first two letters of the last name
- converts all the letters to upper case
- generates a random whole number between 1 and 1000 inclusive.

For the following example:

The first name is `Muhammed`

The last name is `Hann`

The generated number is `492`

The username will be `MHA492`

**(a)** Write the pseudocode to generate a whole random number between 1 and 1000 inclusive and store it in the variable `GeneratedNumber`

.....

.....

..... [2]

**(b)** The last name is input into the variable `LastName`

Write the pseudocode to:

- take the first two letters from `LastName`
- convert the letters to upper case
- store the result in the variable `Temp`

.....

.....

..... [3]

**(c)** The new username is stored in the variable `NewUsername`

Write the pseudocode to store `NewUsername` to the file `Username.txt`

.....

.....

.....

..... [3]

© Cambridge University Press & Assessment 2026 

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

**(d)** Identify and describe verification checks that could be applied to the inputs of the first name and the last name of the user.

.....

.....

.....

.....

.....

..... [4]

**6** Give programming examples for a count-controlled loop and a post-condition loop. Describe each of the loops.

count-controlled loop example .....

description .....

.....

.....

post-condition loop example .....

description .....

.....

.....

[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

**7** An algorithm has been written in pseudocode for a user to repeatedly input positive whole numbers and store them in an array called `Values[]` until –1 is input or until the array is full. The algorithm outputs the average of the numbers to 2 decimal places.

```
01  DECLARE Counter : INTEGER
02  DECLARE Total : INTEGER
03  DECLARE Average : INTEGER
04  DECLARE Values : ARRAY[1:100] OF INTEGER
05  Counter ← 1
06  Total ← 0
07  OUTPUT "Please enter a positive whole number or -1 "
08  INPUT Number
09  WHILE Number = -1 AND Counter <= 100 DO
10      Values[Counter] ← Number
11      Total ← Total + 1
12      Counter ← Counter + 1
13      INPUT Number
14  ENDWHILE
15  Counter ← Counter - 1
16  Average ← Counter / Total
17  OUTPUT "The average is ", ROUND(Average, 2)
```

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 .....

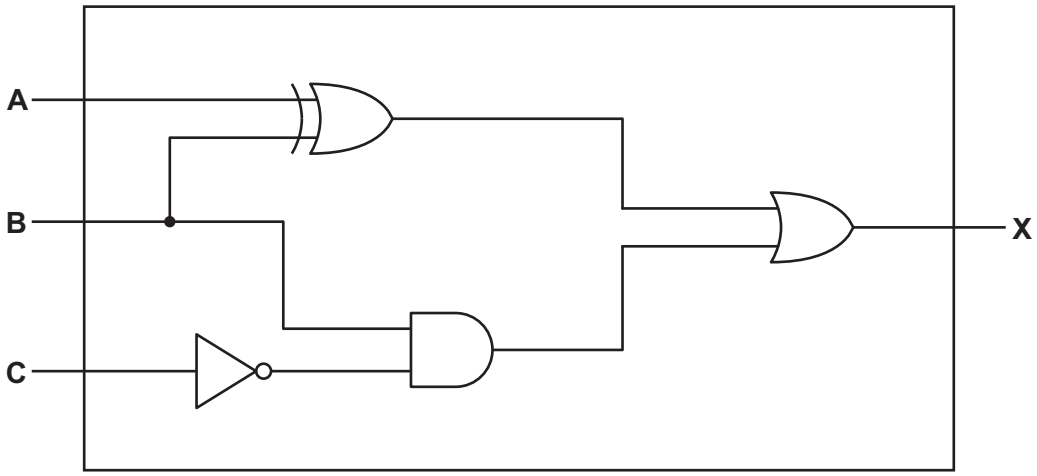
correction .....

.....

[4]

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** Consider the logic circuit:




**(a)** Write the logic expression for the given logic circuit.

.....

..... [3]

**(b)** Complete the truth table for the given logic circuit.

| 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

**9** A swimming club wants to set up a new database table to store details of the members of the club and the maximum distance each member swam in metres.

The table is called `SWIMMERS` and will contain these fields:

- `Distance` – distance the member swam, for example 1500
- `Date` – date the member swam the distance
- `Time` – time taken to swim the distance
- `MemberID` – the code to identify a member, for example TH749
- `MemberName` – name of the member.

**(a) (i)** State why `MemberName` would **not** be appropriate for a primary key.

.....

..... [1]

**(ii)** Give the name of the most appropriate field for the primary key.

.....

..... [1]

**(b)** Complete the table to identify the most appropriate data type for these fields in the table `SWIMMERS`. Each data type must be different.

| field | data type |
|---|---|
| `Distance` | |
| `Date` | |
| `MemberID` | |

[2]

**(c)** Complete the structured query language (SQL) statement to display only the names of the members who swam a distance of 1500 metres or more.

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

[3]

© Cambridge University Press & Assessment 2026 

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

**Turn over**

**10 (a)** Draw a flowchart to input six whole numbers and output the highest. You do **not** need to validate the inputs.

START

[6]

DO NOT WRITE IN THIS MARGIN

© Cambridge University Press & Assessment 2026 

**(b)** Describe how the flowchart could be changed to only accept positive whole numbers.

.....

.....

.....

.....

.....

..... [3]

**11** The function `Power(FirstNumber, SecondNumber)` returns the value of `FirstNumber` raised to the power of `SecondNumber`. `FirstNumber` and `SecondNumber` are integers.

Write pseudocode statements to:

- define the function
- call the function with the variables `A` and `B` and store the return value in the variable `Value`

function definition

.....

.....

.....

.....

.....

.....

function call

.....

.....

[5]

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** A weather monitoring station records the number of minutes of sunshine for each day of a month.

The one-dimensional (1D) array `Sunshine[]` is used to store the number of minutes of sunshine for each day. The index of the array represents the day.

The one-dimensional (1D) array `MonthDays[]` contains the number of days in each month. The index of the array represents the month. In this array, 28 is stored in index 2 for the month of February. However, there are 29 days in February in a leap year.

In this program, a leap year is a year that divides by 4 with no remainder. For example, 2026 is **not** a leap year but 2028 will be a leap year.

Write a program that meets the following requirements:

- allows the user to input a year, calculates whether it is a leap year and updates the number of days in February as appropriate
- allows a user to input the name of one of the first three months of the year: January, February or March
- allows a user to input the number of minutes of sunshine for each day of the month and validates the input
- calculates the total amount of sunshine for the month in hours and minutes
- finds the highest amount of daily sunshine for the month
- outputs the:
  - year
  - month
  - total amount of sunshine for the month in hours and minutes
  - the highest amount of daily sunshine for the month.

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

Do **not** declare any arrays or variables; assume that this has already been done.

All inputs and outputs must contain suitable messages.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

DO NOT WRITE IN THIS MARGIN

© Cambridge University Press & Assessment 2026 

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

© Cambridge University Press & Assessment 2026 







**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 (Cambridge University Press & Assessment) 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 our 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 International Education is the name of our awarding body and a part of Cambridge University Press & Assessment, which is a department of the University of Cambridge.

© Cambridge University Press & Assessment 2026 

