CAMBRIDGE
International Education

Cambridge O Level

COMPUTER SCIENCE **2210/21**

Paper 2 Algorithms, Programming and Logic **May/June 2025**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the May/June 2025 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

This document consists of **19** printed pages.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Annotations guidance for centres

Examiners use a system of annotations as a shorthand for communicating their marking decisions to one another. Examiners are trained during the standardisation process on how and when to use annotations. The purpose of annotations is to inform the standard isation and monitoring processes and guide the supervising examiners when they are checking the work of examiners within their team. The meaning of annotations and how they are used is specific to each component and is understood by all examiners who mark the component.

We publish annotations in our mark schemes to help centres understand the annotations they may see on copies of scripts. Note that there may not be a direct correlation between the number of annotations on a script and the mark awarded. Similarly, the use of an annotation may not be an indication of the quality of the response.

The annotations listed below were available to examiners marking this component in this series.

Annotations

Annotation	Meaning
✓	Correct point
✗	Incorrect point
FT	Follow through
REP	Repetition
I	Ignore
BOD	Benefit of doubt given
TV	Content of response too vague
NAQ	Not answered question
λ	Omission
≀	Section not relevant

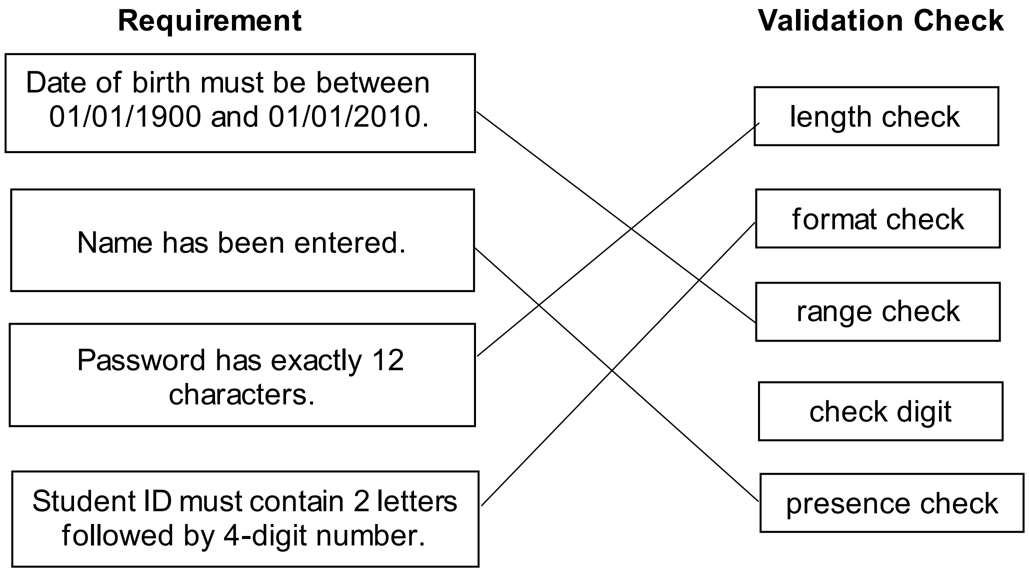
Annotation	Meaning
〰	Section incorrect
Highlighter	Highlights part of the answer or shows structure of complex answers
SEEN	Page or response seen by examiner
A2	AO2 mark
A3	AO3 mark
NE	Not enough
R1	Required item one
R2	Required item two
R3	Required item three
✓ 1	Correct awarding one mark
✓ 2	Correct awarding two marks
✓ 3	Correct awarding three marks
✓ 4	Correct awarding four marks
✓ 5	Correct awarding five marks
✓ 6	Correct awarding six marks
✓ 7	Correct awarding seven marks
✓ 8	Correct awarding eight marks
✓ 9	Correct awarding nine marks

Mark scheme abbreviations

/	separates alternative words / phrases within a marking point
//	separates alternative answers within a marking point
<u>underline</u>	actual word given must be used by candidate (grammatical variants accepted)
max	indicates the maximum number of marks that can be awarded
( )	the word / phrase in brackets is not required, but sets the context

Note: No marks are awarded for using brand names of software packages or hardware.

Question	Answer	Marks
1	C	**1**

Question	Answer	Marks
2(a)	**One** mark for each correct line.  Requirement Validation Check Date of birth must be between 01/01/1900 and 01/01/2010. Name has been entered. Password has exactly 12 characters. Student ID must contain 2 letters followed by 4-digit number. length check format check range check check digit presence check	**4**
2(b)	**One** mark for check, **two** marks for description. • Visual check [1] … • …look at the data that has been entered [1] and confirm it matches [1] • Double entry check [1] … • …enter data twice [1] and only accept identical values [1]	**3**

Question	Answer	Marks

3	**One** mark per bullet point
	• Initialising a count variable outside of loop
	• Correct use of loop with condition for 10 times (`WHILE (DO)...ENDWHILE, REPEAT...UNTIL`) with close
	• Incrementing the count variable inside the loop
	• Rest of algorithm correct
	Examples
	`Count ← 1`
	`REPEAT`
	`    OUTPUT "Please enter a name"`
	`    INPUT Name`
	`    Names[Count] ← Name`
	`    Count ← Count + 1`
	`UNTIL Count > 10`
	Or
	`Count ← 1`
	`WHILE Count <= 10 DO`
	`    OUTPUT "Please enter a name"`
	`    INPUT Name`
	`    Names[Count] ← Name`
	`    Count ← Count + 1`
	`ENDWHILE`
		4

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>4(a)</td><td>

One mark per bullet point. **Max** 5

- Correct input of number
- Use of substring ( )
- …to extract the correct number
- Correct use of IF statements/Case statement with matching `END IF` / `END CASE`
- Output of matching locations
- Output of unknown

Example

```
OUTPUT "Please enter a telephone number "
INPUT Telephone
Value ← SUBSTRING(Telephone, 2, 2)
IF Value = "44"
  THEN
    OUTPUT "UK"
  ELSE
    IF Value = "20"
      THEN
        OUTPUT "Egypt"
      ELSE
        OUTPUT "Unknown"
    ENDIF
ENDIF
```

</td><td>5</td></tr>
<tr><td>4(b)</td><td>

One mark per bullet point. **Max** 3

- Find length of input.
- Use a suitable loop (repeat/while).
- Using the correct condition
- Prompt for a 13-digit telephone number and input number inside loop.

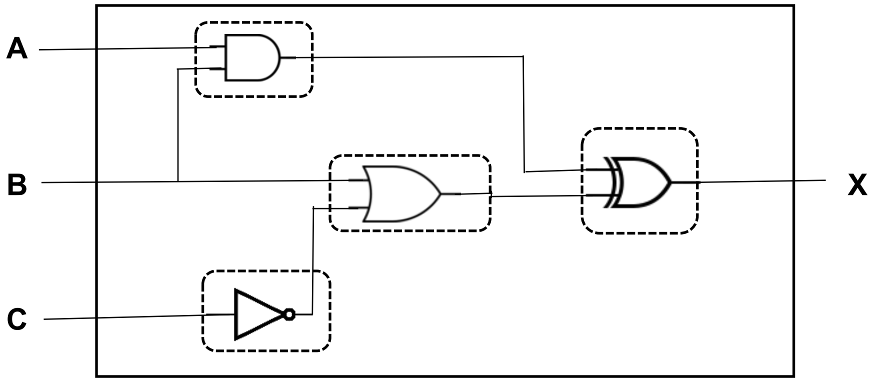
</td><td>3</td></tr>
</table>

Question	Answer	Marks
5	**One** mark for each bullet point • MOD – returns the remainder of a division calculation. • Example: `X ← MOD(10, 3)` • DIV – Integer division/returns the quotient of a division. • Example: `Y ← DIV(10, 3)`	**4**

Question	Answer	Marks
6(a)	**One** mark for each line identified and corrected. Line 02 – should be `N ← ROUND(RANDOM() * 99, 0) + 1` Line 12 – should be `OUTPUT "The number is higher than your guess"` Line 14/15 - `Counter ← Counter + 1` should be placed after Line 15/After `ENDIF`/before `UNTIL G = N`/before Line 16 Line 17 – should be `OUTPUT "Well done, you took ", Counter, " attempts"`	**4**
6(b)	**One** mark for method, **one** mark for example – **max** 2. • Meaningful identifiers, `N` should be `Number`. • Comments, `// loops until the guess equals the number` • Procedures/functions, making the algorithm a procedure/function	**2**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>7(a)</td><td>

One mark for each correct gate, with the correct input(s) as shown.



</td><td>4</td></tr>
<tr><td>7(b)</td><td>

<table>
<tr><th>A</th><th>B</th><th>C</th><th>X</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>**1**</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>**0**</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>**1**</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>**1**</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>**1**</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>**0**</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>**0**</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>**0**</td></tr>
</table>

4 marks for 8 correct outputs
3 marks for 6/7 correct outputs
2 marks for 4/5 correct outputs
1 mark for 2/3 correct outputs

</td><td>4</td></tr>
</table>

Question	Answer	Marks
8(a)(i)	Boolean	**1**
8(a)(ii)	There are **only** 2 responses / **Only** Yes and No / True and False	**1**
8(b)	**Two** marks for all three correct **One** mark for one/two correct BC02 The Fountain 9:10 BC08 Sunset Quay 12:01 BC27 The University 16:45	**2**

8(c)	**One** mark per correct statement
	
	`SELECT Destination`
	`FROM BUS`
	`WHERE Return = Yes;`
		3

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(a)</td><td>

One mark for each correct line

```
DECLARE A: INTEGER
DECLARE B: STRING
```

</td><td>2</td></tr>
<tr><td>9(b)</td><td>

One mark for using `FUNCTION` and `ENDFUNCTION` and `RETURNS STRING`
One mark for naming the function Odds.
One mark for defining the parameter correctly.
One mark for determining if odd/even using `MOD`/`DIV`/`ROUND` using parameter.
One mark for correctly returning ‘Odd’ and ‘Even’.
One mark for correct function call

Example:

```
FUNCTION Odds(X: INTEGER) RETURNS STRING
  IF MOD(X, 2) = 0
    THEN
      RETURN "Even"
    ELSE
      RETURN "Odd"
  ENDIF
ENDFUNCTION

B ← Odds(A)
```

</td><td>6</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>10(a)</td><td>

One mark for each correct column

<table>
<tr><th>Score</th><th>Grade</th><th>OUTPUT</th></tr>
<tr><td>21</td><td>D</td><td>You achieved a D</td></tr>
<tr><td>46</td><td>C</td><td>You achieved a C</td></tr>
<tr><td>63</td><td>C</td><td>You achieved a C</td></tr>
<tr><td>91</td><td>C</td><td>You achieved a C</td></tr>
<tr><td>12</td><td>D</td><td>You achieved a D</td></tr>
</table>

Candidates can be awarded all three marks if they have provided only the correct first row.

</td><td>3</td></tr>
<tr><td>10(b)</td><td>

One mark per bullet point. Max **four**

- Change the decision in the first decision box to IS Score >= 80 ?
- Change the decision in the last decision box to IS Score >= 40 ?
- Change process box from Grade ← "C" to Grade ← "A"
- Change process box from Grade ← "A" to Grade ← "C"
- Add output box instructing users to input score or input −1 before the input box // initialising a counter // incrementing counter.
- Add decision box for loop with exit condition.
- Correct arrows to show looping

</td><td>4</td></tr>
</table>

Question	Answer	Marks
11	Marks are available for: • AO2 (maximum 9 marks) • AO3 (maximum 6 marks) **Data Structures required** names shown underlined must be used as given in the scenario Arrays or lists `CompetitorName`, `CompetitorScore`, `Points` **Requirements (techniques)** R1 inputs and validates the points for each event for each competitor (nested iteration, iteration, input, output) R2 finds and outputs the name of the competitors with the highest score for each event (nested iteration, output and selection) R3 totals the score for each competitor, finds and outputs the names with the highest score for all five events(iteration, totalling, output and selection)	**15**

Question	Answer	Marks
11	**Example 15-mark answer in pseudocode.**	**15**

// meaningful identifier names and appropriate data structures (variables, constants and the given arrays) to store all the data required. Note that candidates did **not** need to declare any arrays or variables as stated in the question paper.

```
DECLARE Points : ARRAY [1 : 25] OF REAL
DECLARE Highest : ARRAY [1 : 5] OF REAL

Winner ← 0
Max ← 0

FOR Count ← 1 TO 25  //initialising array
  Points[Count] ← 0
NEXT Count

FOR Count ← 1 TO 5  //initialising array
  Highest[Count] ← 0
NEXT Count
```

Requirement 1

```
FOR Event ← 1 TO 5
    For X ← 1 TO 25   // loops through each entrant
        REPEAT
            OUTPUT "Please enter a score between 0 and 100 for ", Competitor[X], " for
event ", Event
            INPUT Score
        UNTIL Score >= 0 AND Score <= 100 //validates score
        CompetitorScore[X, Event] ← Score  //stores score for each Event
        Points[X] ← Points[X] + Score  //can total score here
    NEXT X
NEXT Event
```

Question	Answer	Marks
11	Requirement 2	**15**

```
FOR Event ← 1 TO 5
    For X ← 1 TO 25
        IF CompetitorScore[X, Event] > Highest[Event] //calculates the highest score
          THEN
            Highest[Event] = CompetitorScore[X, Event]
        ENDIF
    NEXT X
NEXT Event

FOR Event ← 1 TO 5   //outputs the names of competitors with the highest score
    For X ← 1 TO 25
        OUTPUT "The winner of a medal for the highest score in event ", Event
        IF CompetitorScore[X, Event] = Highest[Event]
          THEN
            OUTPUT CompetitorName[X]
        ENDIF
    NEXT X
NEXT Event
```

Requirement 3

```
//points may be calculated here
For X ← 1 TO 25   //calculates the highest points scored for the five events
    IF Points[X] > Max
      THEN
        Max ← Points[X]
    ENDIF
NEXT X
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>11</td><td>
<pre>
//outputs the names of the competitors with the highest score
OUTPUT "The competitors with the highest score for the five events"
For X ← 1 TO 25
    IF Points[X] = Max
      THEN
        OUTPUT CompetitorName[X]
    ENDIF
NEXT X
</pre>
<table>
<tr><th colspan="4">Marking Instructions in italics</th></tr>
<tr><th colspan="4">AO2: Apply knowledge and understanding of the principles and concepts of computer science to a given context, including the analysis and design of computational or programming problems</th></tr>
<tr><th>0</th><th>1–3</th><th>4–6</th><th>7–9</th></tr>
<tr><td rowspan="2">No creditable response.</td><td>At least one programming technique has been used.
Any use of selection, iteration, counting, totalling, input and output.</td><td>Some programming techniques used are appropriate to the problem.
More than one technique seen applied to the scenario, check the list of techniques needed.</td><td>The range of programming techniques used is appropriate to the problem.
All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check the list of techniques needed.</td></tr>
<tr><td>Some data has been stored but not appropriately.
Any <b>use</b> of variables or arrays or other language dependent data structures e.g. Python lists.</td><td>Some of the data structures chosen are appropriate and store some of the data required.
More than one data structure <b>used</b> to store data required by the scenario.</td><td>The data structures chosen are appropriate and store all the data required.
The data structures <b>used</b> store all the data required by the scenario.</td></tr>
</table>
</td><td><b>15</b></td></tr>
</table>

Question	Answer	Marks
11	**Marking Instructions in italics**	**15**

AO3: Provide solutions to problems by:

- **evaluating computer systems**
- **making reasoned judgements**
- **presenting conclusions**

0	1–2	3–4	5–6
No creditable response.	Program seen without relevant comments.	Program seen with some relevant comment(s).	The program has been fully commented.
	Some identifier names used are appropriate. Some of the data structures used have meaningful names.	The majority of identifiers used are appropriately named. Most of the data structures used have meaningful names.	Suitable identifiers with names meaningful to their purpose have been used throughout. All the data structures used have meaningful names.
	The solution is illogical.	The solution contains parts that may be illogical.	The program is in a logical order.
	The solution is inaccurate in many places. Solution contains few lines of code with errors that attempt to perform a task given in the scenario.	The solution contains parts that are inaccurate. Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.	The solution is accurate. Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.
	The solution attempts at least one of the requirements. Solution contains lines of code that attempt at least one task given in the scenario.	The solution meets most of the requirements. Solution contains lines of code that perform most tasks given in the scenario.	The solution meets all the requirements given in the question. Solution performs all the tasks given in the scenario.