CAMBRIDGE
International Education

Cambridge IGCSE™

COMPUTER SCIENCE **0478/21**

Paper 2 Algorithms, Programming and Logic **October/November 2024**

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 October/November 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

© Cambridge University Press & Assessment 2024

PUBLISHED

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.

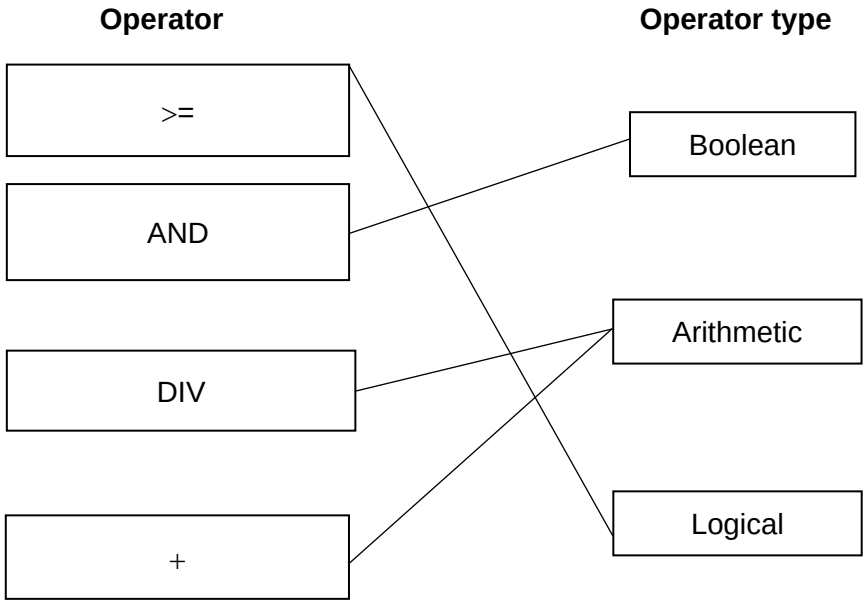
© Cambridge University Press & Assessment 2024

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.

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
1	**D**	**1**

Question	Answer	Marks
2	**C**	**1**

Question	Answer	Marks
3	**One** mark for each correct line  	**4**

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
4	**One** mark for each point • analysis • design • testing	**3**

Question	Answer	Marks
5	**One** mark for each method identified, **one** mark for a further description (**max six)** • structure diagram (1) a hierarchical diagram showing the breakdown of a computer program into sub-programs (1) • flowchart (1) a diagram showing the ordered steps to complete a computer program (1) • pseudocode (1) shows what a program does in plain language (1)	**6**

Question	Answer	Marks
6(a)	**One** mark for each point (**max three)** • `06 T ← 0` • `08 IF A[C] = 0` • `14 NEXT C // NEXT`	**3**

6(b)	**One** mark for outputting `T`
	One mark for outputting `I`
	One mark for suitable messages
	
	Example:
	`15 OUTPUT "Number of zeros in the array is ", T`
	`16 OUTPUT "Total of non-zero elements in the array is ", I`
		3

© Cambridge University Press & Assessment 2024

Question	Answer	Marks

6(c)	**One** mark for a meaningful identifier for the array
	`A MyArray`
	
	One mark for 2 meaningful identifiers, **Two** marks for 3 meaningful identifiers:
	`T Counter`
	`C Index`
	`I Total`
		3

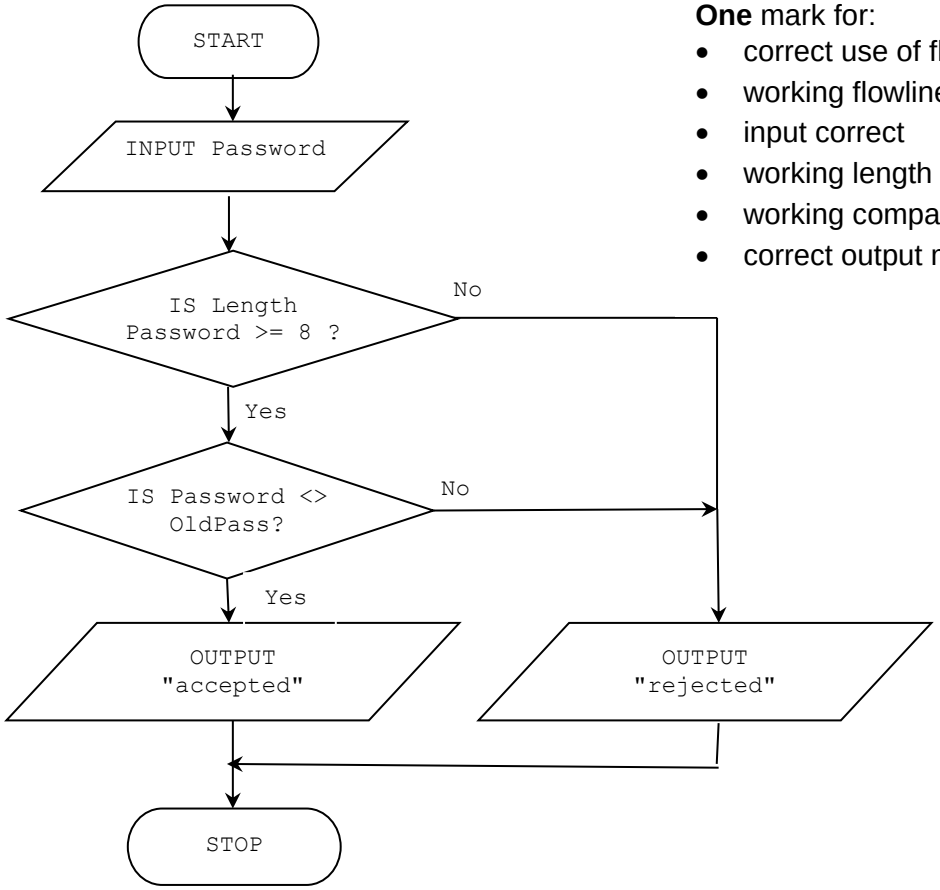
Question	Answer	Marks
7(a)	**One** mark for each point • NOT **X** • **Y** XOR **Z** • expression correct: NOT **X** OR (**Y** XOR **Z**) // (**Y** XOR **Z**) OR NOT **X**	**3**

© Cambridge University Press & Assessment 2024

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

<table>
<tr><th>X</th><th>Y</th><th>Z</th><th>W</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>1</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>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td></tr>
</table></td><td>4</td></tr>
</table>

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
8(a)	 **One** mark for: • correct use of flowchart symbols • working flowlines • input correct • working length check • working comparison • correct output messages	**6**
8(b)	**One** mark for each point • `OPENFILE MyPassword.txt FOR WRITE` • `WRITEFILE MyPassword.txt, Password` • `CLOSEFILE MyPassword.txt`	**3**

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
8(c)	**One** mark for each point • needs to be retrieved on demand // saved for a later date • storage must be non-volatile	**2**

Question	Answer	Marks

9(a)	**One** mark for each error identified and correction:
	• Line `03 Temp should be Temperature`
	• Line `04 = should be <>`
	• Line `14 OR should be AND`
	• Line `19 WHILE should be UNTIL`
	
	`01 REPEAT`
	`02  OUTPUT "Please enter temperature "`
	`03  INPUT Temperature`
	`04  IF Temperature <> 999`
	`05  THEN`
	`06  IF Temperature > 38.0`
	`07  THEN`
	`08  OUTPUT "Temperature too high"`
	`09  ENDIF`
	`10  IF Temperature < 35.0`
	`11  THEN`
	`12  OUTPUT "Temperature too low"`
	`13  ENDIF`
	`14  IF Temperature >= 35.0 AND Temperature <= 38.0`
	`15  THEN`
	`16  OUTPUT "Temperature normal"`
	`17  ENDIF`
	`18  ENDIF`
	`19 UNTIL Temperature = 999`
		4

© Cambridge University Press & Assessment 2024

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>9(b)</td><td>One mark for each point
• greater than or equal to 35
• and less than or equal to 38</td><td>2</td></tr>
<tr><td>9(c)</td><td>One mark for each correct column max two
<table>
<tr><th>Temperature</th><th>OUTPUT</th></tr>
<tr><td></td><td>(Please enter temperature)</td></tr>
<tr><td>34.22</td><td>Temperature too low</td></tr>
<tr><td></td><td>(Please enter temperature)</td></tr>
<tr><td>36.1</td><td>Temperature normal</td></tr>
<tr><td></td><td>(Please enter temperature)</td></tr>
<tr><td>37.4</td><td>Temperature normal</td></tr>
<tr><td></td><td>(Please enter temperature)</td></tr>
<tr><td>38.0</td><td>Temperature normal</td></tr>
<tr><td></td><td>(Please enter temperature)</td></tr>
<tr><td>999</td><td></td></tr>
</table></td><td>2</td></tr>
</table>

Question	Answer	Marks
10(a)	19	**1**
10(b)(i)	ChNo	**1**

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
10(b)(ii)	Unique identifier	**1**
10(c)	**One** mark for each • `CH04 0.0` • `CH05 50.0`	**2**

10(d)(i)	**One** mark each correct line
	`SELECT Name`
	`FROM CheeseStock`
	`WHERE NOT InStock // InStock = No / "No" / False // WHERE WeightKg = 0.0;`
		3

10(d)(ii)	**One** mark for each point • `WHERE clause can check either the InStock field for FALSE / No / "No" or` • `the WeightKg field for 0.0`	**2**

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
11	• AO2 (maximum 9 marks) • AO3 (maximum 6 marks) **Data Structures required** names shown underlined must be used as given in the scenario. 1D Array or list `MemberName[], MemberTime[], MemberCertificate[], Position[]` Variables `Index, Last, ClubSize, TempTime, TempName, Swap, Count` **Requirements (techniques)** **R1** Input and verify the members times (input and iteration) **R2** sort the MemberName[] and MemberTime[] arrays in ascending order of time and outputs the top three members and their times (nested iteration, sorting, selection and output) **R3** Storing the members names who will receive a certificate and outputting the number of certificates (iteration, selection, counting and output)	**15**

© Cambridge University Press & Assessment 2024

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

```
CONSTANT ClubSize = 200  // setting the number of members in the club
DECLARE Array Position[1:3} STRING
DECLARE Position : ARRAY[1:3] OF STRING

Position[1] ← "First"
Position[2] ← "Second"
Position[3] ← "Third"

FOR Index ← 1 TO ClubSize // 200
    REPEAT
        PRINT "Please enter the time for ", MemberName[Index]
        INPUT Time1
        PRINT "Please re-enter the time"
        INPUT Time2
        IF Time1 <> Time2
          THEN
            PRINT "Incorrect input, the times should be the same, please re-enter"
        ENDIF
    UNTIL Time1 = Time2
    MemberTime[Index] ← Time1
NEXT Index

Last ←ClubSize
REPEAT
    Swap ← FALSE
    FOR Index ← 1 TO ClubSize - 1
        IF MemberTime[Index] > MemberTime[Index + 1]
          THEN
            TempTime ← MemberTime[Index]
            MemberTime[Index] ← MemberTime[Index + 1]
            MemberTime[Index + 1] ← TempTime
            TempName ← MemberName[Index]
            MemberName[Index] ← MemberName[Index + 1]
            MemberName[Index + 1] ← TempName
```

© Cambridge University Press & Assessment 2024

Question	Answer	Marks
11	Swap ← TRUE ENDIF NEXT Index Last ← Last - 1 UNTIL NOT Swap or Last = 1 FOR Index ← 1 TO 3 OUTPUT Position[Index], MemberName[Index], " with a time of ", MemberTime[Index] NEXT Index Count ← 0 FOR Index ← 1 TO ClubSize // 200 IF MemberTime[Index] < 240 THEN Count ← Count + 1 MemberCertificate[Count] ← MemberName[Index] ENDIF NEXT Index OUTPUT "Number of certificates to be printed is ", Count	

© Cambridge University Press & Assessment 2024

Marking Instructions in italics			
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			
0	**1–3**	**4–6**	**7–9**
No creditable response.	At least one programming technique has been used. *Any use of selection, iteration, counting, totalling, input and output.*	Some programming techniques used are appropriate to the problem. *More than one technique seen applied to the scenario, check list of techniques needed.*	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 list of techniques needed.*
	Some data has been stored but not appropriately. *Any **use** of variables or arrays or other language dependent data structures e.g. Python lists.*	Some of the data structures chosen are appropriate and store some of the data required. *More than one data structure **used** to store data required by the scenario.*	The data structures chosen are appropriate and store all the data required. *The data structures **used** store all the data required by the scenario.*

© Cambridge University Press & Assessment 2024

Marking Instructions in italics			
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 of 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.*

© Cambridge University Press & Assessment 2024