



--

--	--	--	--	--

--	--	--	--

0478/23

October/November 2019

**1 hour 45 minutes**

No Additional Materials are required.

No calculators allowed.

## READ THESE INSTRUCTIONS FIRST

Write in dark blue or black pen.

You may use an HB pencil for any diagrams, graphs or rough working.

Do not use staples, paper clips, glue or correction fluid.

DO **NOT** WRITE IN ANY BARCODES.

Answer **all** questions.

**DO NOT ATTEMPT TASKS 1, 2 AND 3** in the pre-release material; these are for information only.

You are advised to spend no more than **40 minutes** on **Section A** (Question 1).

No marks will be awarded for using brand names of software packages or hardware.

Any businesses described in this paper are entirely fictitious.

At the end of the examination, fasten all your work securely together.

The number of marks is given in brackets [ ] at the end of each question or part question.

The maximum number of marks is 50.

This syllabus is regulated for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

This document consists of **12** printed pages.

## Section A

You are advised to spend no longer than 40 minutes answering this section.

Here is a copy of the pre-release material.

**DO NOT** attempt Tasks 1, 2 and 3 now.

Use the pre-release material and your experience from attempting the tasks before the examination to answer Question 1.

## Pre-release material

You have been asked to write a program to calculate the area of a wall and the cost of the tiles needed to cover it. The program should work for any room with up to four walls being tiled. Tiles are sold in boxes; each box covers one square metre.

Tile description	Price per box
Small black granite	\$19.50
Small grey marble	\$25.95
Small powder blue	\$35.75
Medium sunset yellow	\$12.50
Medium berry red	\$11.00
Medium glitter purple	\$52.95
Large oak wood effect	\$65.00
Large black granite	\$58.98
Large bamboo effect	\$85.00
Extra-large white marble	\$62.75

Write and test a program or programs to calculate the area of the walls and the cost of tiles needed to tile a room.

- Your program or programs must include appropriate prompts for the entry of data; data must be validated on entry.
- Error messages and other output need to be set out clearly and understandably.
- All arrays, variables, constants and other identifiers must have meaningful names.

You will need to complete these **three** tasks. Each task must be fully tested.

**Task 1** – Setting up and using the system for a single wall.

Set up your program to:

- Store the tile description, price per box and an identification code using a suitable programming technique.
- Display the list of tiles including the identification code, description and price per box.
- Prompt the user to input the height and width of the wall, and the identification code of the tiles they would like.
- Calculate and display the area of the wall, the number of boxes of tiles required (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

**Task 2** – Adding additional walls.

Extend your program to:

- Enter the number of walls to be tiled and the identification code of the tiles to be used. Only one identification code to be used for all walls.
- Enter the dimensions for each wall to be tiled.
- Calculate and display the total area of all walls, the number of boxes of tiles required (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

**Task 3** – Allowing for waste.

It is likely that some of the tiles will not be useable so it is sensible to allow a percentage for wastage, for example 10%. Alter your program to allow the user to input a percentage to calculate wastage and add this to the total area to be tiled. Calculate and display the revised number of boxes of tiles to be purchased (tiles are sold in boxes of complete square metres) and the total cost of the tiles.

- 1 (a) All variables, constants and other identifiers must have meaningful names.

Name **two** arrays you could use for **Task 1**. State the data type and purpose of each one.

Array 1 .....

Data type 1 .....

Purpose .....

.....

Array 2 .....

Data type 2 .....

Purpose .....

.....

[4]

- (b) Name a variable that you used for **Task 2**. State the data type and purpose of this variable. Explain why you chose to use a variable rather than a constant.

Variable .....

Data type .....

Purpose .....

.....

.....

Reason for use of a variable .....

.....

.....

[4]

- (c) Write an algorithm for **Task 1**, using **either** pseudocode, programming statements **or** a flowchart. Assume that the first part of **Task 1**, the storage and display of tile descriptions, prices per box and identification codes has already been done.

[illegible]



- (d) Explain how you extended your program to complete **Task 2**. Any programming statements you use in your answer must be fully explained.

[3]

(e) Explain how you could validate the input for percentage waste allowance in **Task 3**.

.....

.....

.....

.....

.....

.....

.....

.....

..... [3]

**Section B**

- 2** Describe the use of a subroutine in a program.

.....

.....

.....

.....

..... [2]

- 3** Name the **three** types of loop structure used in pseudocode.

.....

.....

.....

.....

.....

..... [3]



4 The following pseudocode algorithm uses nested IF statements.

```

IF Response = 1
  THEN
    X ← X + Y
  ELSE
    IF Response = 2
      THEN
        X ← X - Y
      ELSE
        IF Response = 3
          THEN
            X ← X * Y
          ELSE
            IF Response = 4
              THEN
                X ← X / Y
              ELSE
                OUTPUT "No response"
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDIF

```

(a) Name the type of statement demonstrated by the use of IF ... THEN ... ELSE ... ENDIF

.....  
 ..... [1]

(b) Re-write the pseudocode algorithm using a CASE statement.

.....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 .....  
 ..... [4]

5 The algorithm performs an operation on the array named `MyData`

`DIV` means integer division, so only the whole number part of the result is returned  
e.g. `7 DIV 2` returns a value of 3

```

First ← 0
Last ← 16
Found ← FALSE
INPUT UserIn
WHILE (First ≤ Last) AND (Found = FALSE) DO
    Middle ← (First + Last) DIV 2
    IF MyData[Middle] = UserIn
    THEN
        Found ← TRUE
    ELSE
        IF UserIn < MyData[Middle]
        THEN
            Last ← Middle - 1
        ELSE
            First ← Middle + 1
        ENDIF
    ENDIF
ENDIF
ENDWHILE
OUTPUT Found

```

This table shows the contents of the array: `MyData` e.g. `MyData[2]` stores the value 5

	<b>MyData</b>																
<b>Index</b>	<b>[0]</b>	<b>[1]</b>	<b>[2]</b>	<b>[3]</b>	<b>[4]</b>	<b>[5]</b>	<b>[6]</b>	<b>[7]</b>	<b>[8]</b>	<b>[9]</b>	<b>[10]</b>	<b>[11]</b>	<b>[12]</b>	<b>[13]</b>	<b>[14]</b>	<b>[15]</b>	<b>[16]</b>
<b>Value</b>	2	3	5	6	8	10	12	13	14	16	18	20	25	27	29	34	36

(a) Complete the trace table for the input data: 10

<b>First</b>	<b>Last</b>	<b>UserIn</b>	<b>Middle</b>	<b>Found</b>	<b>OUTPUT</b>

[6]

(b) Describe the function being performed by the algorithm.

.....

.....

.....

..... [2]

- 6 Draw **four** different flowchart symbols and describe how they are used in a program flowchart.

Flowchart symbol	Description of use

[4]

- 7 A teacher has decided to use a database table as her mark book for her Computer Science class, which she has called MARKBOOK. For each student, the following data will be recorded: first name, last name, their year 10 test score and their year 11 test score. The class has 32 students.

(a) State the number of fields and records required for this database.

Number of Fields .....

Number of Records .....

[2]

(b) The data in MARKBOOK is stored under category headings: LastName, FirstName, Y10TestScore and Y11TestScore.

State, with a reason, whether any of these headings would be suitable as a primary key.

.....

.....

.....

..... [2]

(c) Complete the query-by-example grid to only display the first name, last name and year 10 test score of each student who achieved 50 or more in their year 10 test. The output should be in test score order with the highest marks at the top of the list.

Field:				
Table:				
Sort:				
Show:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Criteria:				
or:				

[4]

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](http://www.cambridgeinternational.org) after the live examination series.

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



**Cambridge Assessment International Education**  
Cambridge International General Certificate of Secondary Education

---

**COMPUTER SCIENCE**

**0478/23**

Paper 2

**October/November 2019**

MARK SCHEME

Maximum Mark: 50

---

**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 2019 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

---

This syllabus is regulated for use in England, Wales and Northern Ireland as a Cambridge International Level 1/Level 2 Certificate.

---

This document consists of **8** 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 descriptors 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.

Question	Answer	Marks
1(a)	<p>Two examples of:</p> <p>Any meaningful name for an array related to <b>Task 1</b> – <b>one</b> mark, e.g.</p> <ul style="list-style-type: none"> <li>• TileDescription</li> <li>• TilePrice</li> <li>• TileCode</li> </ul> <p>Correct data type <b>and</b> purpose related to <b>Task 1</b> – <b>one</b> mark, e.g.</p> <ul style="list-style-type: none"> <li>• ... string to store the name / description of the tiles</li> <li>• ... real to store the price of the tiles</li> <li>• ... string to store the tile codes</li> </ul>	<b>4</b>
1(b)	<p>Any meaningful name for a variable related to <b>Task 2</b> – <b>one</b> mark, e.g.</p> <ul style="list-style-type: none"> <li>• NumberOfWalls</li> <li>• TotalArea</li> <li>• TotalCost</li> </ul> <p>Relevant data type for the variable related to <b>Task 2</b> – <b>one</b> mark, e.g.</p> <ul style="list-style-type: none"> <li>• ... Integer</li> <li>• ... Real</li> </ul> <p>Relevant purpose for the variable related to <b>Task 2</b> – <b>one</b> mark, e.g.</p> <ul style="list-style-type: none"> <li>• ... to store the number of walls that need to be tiled</li> <li>• ... to store the total area / cost</li> </ul> <p><b>One</b> mark for a correct reason, e.g.</p> <ul style="list-style-type: none"> <li>• Variables allow the storage of values within a program that may change as the program runs // Variables are used to store values that are input or calculated</li> </ul>	<b>4</b>

Question	Answer	Marks
1(c)	<p><b>Six from:</b></p> <p>MP1 Input for height and width of the wall to tile and tile code</p> <p>MP2 Prompts for all inputs seen</p> <p>MP3 Validation of height and width</p> <p>MP4 Reasonable attempt at validation of tile code</p> <p>MP5 Calculation of the area of the wall</p> <p>MP6 Calculation of the number of boxes rounded up</p> <p>MP7 Looking up the cost of the tiles</p> <p>MP8 Calculation of the cost of the boxes of the tiles</p> <p>MP9 Output of area of the wall, the number of boxes and cost of tiles needed</p> <p>MP10 Appropriate message(s) with output</p> <p><b>Example algorithm</b></p> <pre> REPEAT     OUTPUT "Please enter height of wall "     INPUT Height UNTIL Height &gt; 0  REPEAT     OUTPUT "Please enter width of wall "     INPUT Width UNTIL Width &gt; 0  REPEAT     Found ← FALSE     Counter ← 0     REPEAT         OUTPUT "Please enter tile identification code "         INPUT MyTileId         IF MyTileId = TileCode(Counter)             THEN                 MyCost ← TilePrice(Counter)                 Found ← TRUE             ENDIF         Counter ← Counter + 1     UNTIL Found OR Counter = 10 UNTIL Found  Area ← Height * Width NoBoxes ← Int(Round (Area + 0.5)) // Rounds up to nearest Integer TotalPrice ← NoBoxes * MyCost  OUTPUT ("Area of wall is ", Area, " metres") OUTPUT ("Number of boxes of tiles is ", Area, " NoBoxes) OUTPUT ("Cost of Tiles is ", TotalPrice, " dollars") </pre>	6



Question	Answer	Marks
1(d)	<p><b>Three</b> from:</p> <ul style="list-style-type: none"> <li>• Adding an extra user input for number of walls ...</li> <li>• ... using this value as a loop counter</li> <li>• ... so that separate inputs of height and width can be made for each wall</li> <li>• Calculating a running total for the final area using the results of the area calculation for each wall</li> <li>• Rounding the final area or using previously rounded areas for the final total</li> <li>• Displaying with a suitable message final area, final number of boxes of tiles required and final cost</li> </ul> <p>If only program statements given with no explanation, <b>zero</b> marks.</p>	<b>3</b>
1(e)	<p><b>Three</b> from:</p> <ul style="list-style-type: none"> <li>• Check that only numbers are accepted // type check</li> <li>• Check that the values are within boundaries // range check</li> <li>• Check that a value has been entered // presence check</li> <li>• Using an IF / conditional statement</li> <li>• Identification of suitable lower value / acceptable value</li> <li>• Identification of suitable upper value / unacceptable value</li> <li>• Alerting the user with an error message if the input is unacceptable</li> </ul>	<b>3</b>



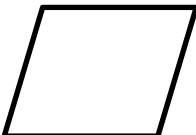
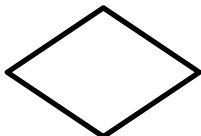
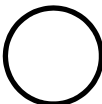


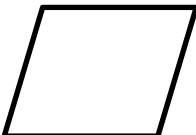
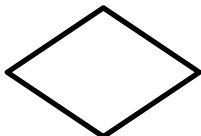
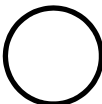


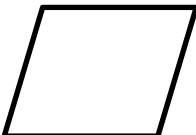
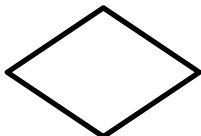
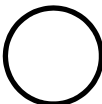
Question	Answer	Marks
2	<b>Two</b> from <ul style="list-style-type: none"> <li>Sub-program / system not the whole program / system</li> <li>To perform a frequently used operation within a program</li> <li>That can be called when needed</li> <li>That can be reused by another program</li> </ul>	2

Question	Answer	Marks
3	<ul style="list-style-type: none"> <li>FOR (... TO ... NEXT) loop</li> <li>WHILE (... DO ... ENDWHILE) loop</li> <li>REPEAT (... UNTIL) loop</li> </ul>	3

Question	Answer	Marks
4(a)	<ul style="list-style-type: none"> <li>Conditional / selection statement</li> </ul>	1
4(b)	<b>Four</b> from: MP1 CASE statement with identifier Response MP2 Correct structure used for choices... MP3 .... correct statements used for choices MP4 OTHERWISE and correct statement MP5 Single ENDCASE included e.g. <pre> CASE OF Response // CASE Response OF   1 : X ← X + Y   2 : X ← X - Y   3 : X ← X * Y   4 : X ← X / Y   OTHERWISE OUTPUT "No response" ENDCASE           </pre>	4

Question	Answer						Marks
5(a)	First	Last	UserIn	Middle	Found	OUTPUT	6
	0	16			FALSE		
	0	16	10	8	FALSE		
	0	7	10	3	FALSE		
	4	7	10	5	TRUE	TRUE	
	One mark per correct column						

Question	Answer	Marks
5(b)	<b>Two</b> from: <ul style="list-style-type: none"> <li>• Search for the value input ...</li> <li>• ... using an array...</li> <li>• ... of sorted data</li> </ul>	<b>2</b>

Question	Answer	Marks												
6	<p><b>One</b> mark for each correct symbol <b>and</b> name / description / example of use (maximum four marks)</p> <table><tr><th>Symbol</th><th>Description of use</th></tr><tr><td></td><td>Terminator – start / end the flowchart</td></tr><tr><td></td><td>Process – to show calculations, etc.</td></tr><tr><td></td><td>Input / Output</td></tr><tr><td></td><td>Decision – to show condition</td></tr><tr><td></td><td>Continuation – to extend the flowchart and allow it to join up</td></tr></table>	Symbol	Description of use		Terminator – start / end the flowchart		Process – to show calculations, etc.		Input / Output		Decision – to show condition		Continuation – to extend the flowchart and allow it to join up	4
Symbol	Description of use													
	Terminator – start / end the flowchart													
	Process – to show calculations, etc.													
	Input / Output													
	Decision – to show condition													
	Continuation – to extend the flowchart and allow it to join up													

Question	Answer	Marks																														
7(a)	<ul style="list-style-type: none"><li>• Number of Fields: 4</li><li>• Number of Records: 32</li></ul>	2																														
7(b)	<ul style="list-style-type: none"><li>• No field is suitable as a primary key ...</li><li>• ... because none of the data would be unique // duplicates could occur</li></ul>	2																														
7(c)	<table><tr><td>Field:</td><td>FirstName</td><td>LastName</td><td>Y10TestScore</td><td></td></tr><tr><td>Table:</td><td>MARKBOOK</td><td>MARKBOOK</td><td>MARKBOOK</td><td></td></tr><tr><td>Sort:</td><td></td><td></td><td>Descending</td><td></td></tr><tr><td>Show:</td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input checked="" type="checkbox"/></td><td><input type="checkbox"/></td></tr><tr><td>Criteria:</td><td></td><td></td><td>&gt;=50</td><td></td></tr><tr><td>or:</td><td></td><td></td><td></td><td></td></tr></table> <p><b>One</b> mark for each completely correct column down to and including ‘Show’ row (maximum three marks) <b>One</b> mark for correct search criteria rows</p>	Field:	FirstName	LastName	Y10TestScore		Table:	MARKBOOK	MARKBOOK	MARKBOOK		Sort:			Descending		Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Criteria:			>=50		or:					4
Field:	FirstName	LastName	Y10TestScore																													
Table:	MARKBOOK	MARKBOOK	MARKBOOK																													
Sort:			Descending																													
Show:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>																												
Criteria:			>=50																													
or:																																