



Cambridge Assessment International Education
Cambridge Ordinary Level

COMPUTER SCIENCE

2210/23

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 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) | Two examples of: Any meaningful name for an array related to Task 1 – one mark, e.g. <ul style="list-style-type: none"> • TileDescription • TilePrice • TileCode Correct data type and purpose related to Task 1 – one mark, e.g. <ul style="list-style-type: none"> • ... string to store the name / description of the tiles • ... real to store the price of the tiles • ... string to store the tile codes | 4 |
| 1(b) | Any meaningful name for a variable related to Task 2 – one mark, e.g. <ul style="list-style-type: none"> • NumberOfWalls • TotalArea • TotalCost Relevant data type for the variable related to Task 2 – one mark, e.g. <ul style="list-style-type: none"> • ... Integer • ... Real Relevant purpose for the variable related to Task 2 – one mark, e.g. <ul style="list-style-type: none"> • ... to store the number of walls that need to be tiled • ... to store the total area / cost One mark for a correct reason, e.g. <ul style="list-style-type: none"> • 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 | 4 |

| Question | Answer | Marks |
|----------|---|-------|
| 1(c) | <p>Six from:</p> <p>MP1 Input for height and width of the wall to tile and tile code MP2 Prompts for all inputs seen MP3 Validation of height and width MP4 Reasonable attempt at validation of tile code MP5 Calculation of the area of the wall MP6 Calculation of the number of boxes rounded up MP7 Looking up the cost of the tiles MP8 Calculation of the cost of the boxes of the tiles MP9 Output of area of the wall, the number of boxes and cost of tiles needed MP10 Appropriate message(s) with output</p> <p>Example algorithm</p> <pre> REPEAT OUTPUT "Please enter height of wall " INPUT Height UNTIL Height > 0 REPEAT OUTPUT "Please enter width of wall " INPUT Width UNTIL Width > 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>Three from:</p> <ul style="list-style-type: none"> • Adding an extra user input for number of walls ... • ... using this value as a loop counter • ... so that separate inputs of height and width can be made for each wall • Calculating a running total for the final area using the results of the area calculation for each wall • Rounding the final area or using previously rounded areas for the final total • Displaying with a suitable message final area, final number of boxes of tiles required and final cost <p>If only program statements given with no explanation, zero marks.</p> | 3 |
| 1(e) | <p>Three from:</p> <ul style="list-style-type: none"> • Check that only numbers are accepted // type check • Check that the values are within boundaries // range check • Check that a value has been entered // presence check • Using an IF / conditional statement • Identification of suitable lower value / acceptable value • Identification of suitable upper value / unacceptable value • Alerting the user with an error message if the input is unacceptable | 3 |



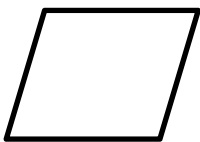
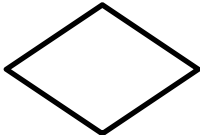
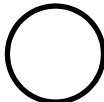


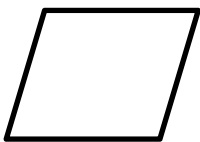
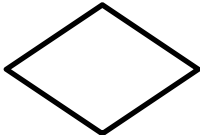
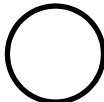


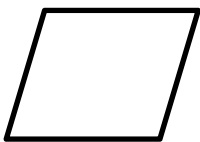
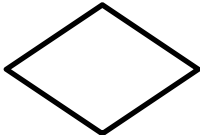
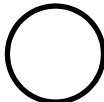
| Question | Answer | Marks |
|----------|---|-------|
| 2 | Two from <ul style="list-style-type: none"> Sub-program / system not the whole program / system To perform a frequently used operation within a program That can be called when needed That can be reused by another program | 2 |

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

| Question | Answer | Marks |
|----------|---|-------|
| 4(a) | <ul style="list-style-type: none"> Conditional / selection statement | 1 |
| 4(b) | Four 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) | <table border="1"> <thead> <tr> <th>First</th> <th>Last</th> <th>UserIn</th> <th>Middle</th> <th>Found</th> <th>OUTPUT</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>16</td> <td></td> <td></td> <td>FALSE</td> <td></td> </tr> <tr> <td>0</td> <td>16</td> <td>10</td> <td>8</td> <td>FALSE</td> <td></td> </tr> <tr> <td>0</td> <td>7</td> <td>10</td> <td>3</td> <td>FALSE</td> <td></td> </tr> <tr> <td>4</td> <td>7</td> <td>10</td> <td>5</td> <td>TRUE</td> <td>TRUE</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>One mark per correct column</p> | First | Last | UserIn | Middle | Found | OUTPUT | 0 | 16 | | | FALSE | | 0 | 16 | 10 | 8 | FALSE | | 0 | 7 | 10 | 3 | FALSE | | 4 | 7 | 10 | 5 | TRUE | TRUE | | | | | | | | | | | | | 6 |
| First | Last | UserIn | Middle | Found | OUTPUT | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 16 | | | FALSE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 16 | 10 | 8 | FALSE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 7 | 10 | 3 | FALSE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 7 | 10 | 5 | TRUE | TRUE | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

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

| Question | Answer | Marks | | | | | | | | | | | | |
|---|---|----------|--------------------|---|--|--|--------------------------------------|---|----------------|---|------------------------------|---|--|--|
| 6 | One mark for each correct symbol and name / description / example of use (maximum four marks) | 4 | | | | | | | | | | | | |
| | <table border="1"> <thead> <tr> <th>Symbol</th> <th>Description of use</th> </tr> </thead> <tbody> <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> </tbody> </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 | |
| 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"> Number of Fields: 4 Number of Records: 32 | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7(b) | <ul style="list-style-type: none"> No field is suitable as a primary key because none of the data would be unique // duplicates could occur | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7(c) | <table border="1" data-bbox="414 450 1348 864"> <tbody> <tr> <td data-bbox="414 450 651 517">Field:</td> <td data-bbox="651 450 884 517">FirstName</td> <td data-bbox="884 450 1118 517">LastName</td> <td data-bbox="1118 450 1348 517">Y10TestScore</td> <td data-bbox="1348 450 1474 517"></td> </tr> <tr> <td data-bbox="414 517 651 584">Table:</td> <td data-bbox="651 517 884 584">MARKBOOK</td> <td data-bbox="884 517 1118 584">MARKBOOK</td> <td data-bbox="1118 517 1348 584">MARKBOOK</td> <td data-bbox="1348 517 1474 584"></td> </tr> <tr> <td data-bbox="414 584 651 651">Sort:</td> <td data-bbox="651 584 884 651"></td> <td data-bbox="884 584 1118 651"></td> <td data-bbox="1118 584 1348 651">Descending</td> <td data-bbox="1348 584 1474 651"></td> </tr> <tr> <td data-bbox="414 651 651 730">Show:</td> <td data-bbox="651 651 884 730"><input checked="" type="checkbox"/></td> <td data-bbox="884 651 1118 730"><input checked="" type="checkbox"/></td> <td data-bbox="1118 651 1348 730"><input checked="" type="checkbox"/></td> <td data-bbox="1348 651 1474 730"><input type="checkbox"/></td> </tr> <tr> <td data-bbox="414 730 651 797">Criteria:</td> <td data-bbox="651 730 884 797"></td> <td data-bbox="884 730 1118 797"></td> <td data-bbox="1118 730 1348 797">>=50</td> <td data-bbox="1348 730 1474 797"></td> </tr> <tr> <td data-bbox="414 797 651 864">or:</td> <td data-bbox="651 797 884 864"></td> <td data-bbox="884 797 1118 864"></td> <td data-bbox="1118 797 1348 864"></td> <td data-bbox="1348 797 1474 864"></td> </tr> </tbody> </table> <p data-bbox="288 898 1334 999"> One mark for each completely correct column down to and including 'Show' row (maximum three marks) One 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: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |