Cambridge International Examinations
Cambridge International General Certificate of Secondary Education

COMPUTER SCIENCE **0478/22**

Paper 2 Algorithms, Programming and Logic **February/March 2025**

MARK SCHEME

Maximum Mark : 75

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.

Cambridge Assessment International Education

© UCLES 2025

[Turn over

Please note the following further points:

Please also read the additional guidance in the mark scheme, it provides further information about how to mark the question.

Please annotate your scripts. The number of ticks given must match the number of marks given.

If you award a benefit of doubt (BOD) mark, this must also have a tick.

The sections in brackets in the mark scheme are not necessary in the candidates answer.

The words in bold in the mark scheme are important text that needs to be present, or some notion of it needs to be present. It doesn't have to be the exact word, but something close to the meaning.

If a word is underlined, this exact word must be present.

A single forward slash means this is an alternative word. A double forward slash means that this is an alternative mark point.

Red text in the mark scheme is a response that we think it just about okay, but that will not be published as a response.

Ellipsis (…) on the end of one mark point and the start of the next means that the candidate cannot get the second mark point without being awarded the first one.

Please inform your team leader when you have submitted your standardisation scripts. They may have many examiners that they are monitoring, so may not see your submission immediately.

The mark scheme has been agreed at standardisation, so it is the AE's responsibility to apply it. You may have differing opinions on how the paper should be marked, but the mark scheme is what has been agreed by a panel and AE's are required to apply it.

Please mark your allocation at a steady rate. If you are not able to mark your allocation for a number of days, inform your team leader, do not wait for them to have to chase you.

Please ensure that you meet the 40% deadline. If you are not close to or meeting this at the 40% deadline, some of your allocation may be reallocated. If you are going to struggle with this deadline, you must inform your team leader ASAP.

There are **3** blank pages at the start of the exam paper. These, along with the flowchart at the start of question 7, **must** be annotated with a **SEEN**, to indicate they have been checked for any further responses. Also, any blank responses or pages must have a **SEEN** annotation, as every question must be annotated. If you do not annotate correctly, you may be stopped by your TL.

If a candidate writes outside the zoned area for the question, this must be linked to the response, even if it is not awarded a mark. This demonstrates at EAR that you did read this part of the response.

If a candidate has crossed out the final section of a response to a question and has not written anything after the crossed-out work, the crossed-out work must be marked.

Cambridge Assessment International Education

© UCLES 2025

[Turn over

Mechanics of Marking:

Every mark given should have a corresponding tick on the script. The number of ticks on each (part) question should match the number of marks awarded for that (part) question. If giving Benefit of Doubt, **the BOD must be accompanied by a tick**.

If a candidate has not given a response or the response is in no way related to the question, such as 'don't know', NR (the Hash key) should be awarded rather than zero.

Every part question must be annotated to show that it has been read even if awarding NR. Please ensure that all part questions that are marked as NR are also annotated with the SEEN icon. This is a requirement of RM3.

There are **3** blank pages at the start of each script and a page with a flowchart at the start of question 7. These must be annotated with the SEEN icon. ☹

NEW: Words or phrases that are underlined, must be present in the candidate's answer. Words or phrases that are emboldened indicate that the idea represented by the bold text must be included.

Even though the comments box is visible at the bottom of the screen, please do not put comments or question marks on the scripts. When scripts are returned to centres all the annotations including comments, are visible.

If work has been crossed out and something written in its place, the replacement work is marked even if the crossed-out work is correct. If the crossed-out work has not been replaced, mark the crossed-out answer. Please also annotate the unmarked work as SEEN, especially if the replacement answer is on a separate sheet.

For single mark answers, mark the first answer on the line, unless there is a note to the contrary on the mark scheme.

If a candidate writes something that is not enough (NE) for a mark, but is not actually incorrect, continue reading, even if the mark scheme says, for example, mark first two answers.

Cambridge Assessment International Education

© UCLES 2025

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 standardisation 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
	Follow through
	Repetition
	Ignore
	Benefit of doubt given
	Benefit of doubt not given
	Content of response too vague
	Not answered question
	Omission

Cambridge Assessment International Education

© UCLES 2025

Annotation	Meaning
	Section not relevant
	Section incorrect
Highlighter	Highlights part of the answer or shows structure of complex answers
	Page or response seen by examiner
	AO2 mark
	AO3 mark
	Not enough
	Required item one
	Required item two
	Required item three
	Correct awarding one mark
	Correct awarding two marks
	Correct awarding three marks
	Correct awarding four marks
	Correct awarding five marks
	Correct awarding six marks

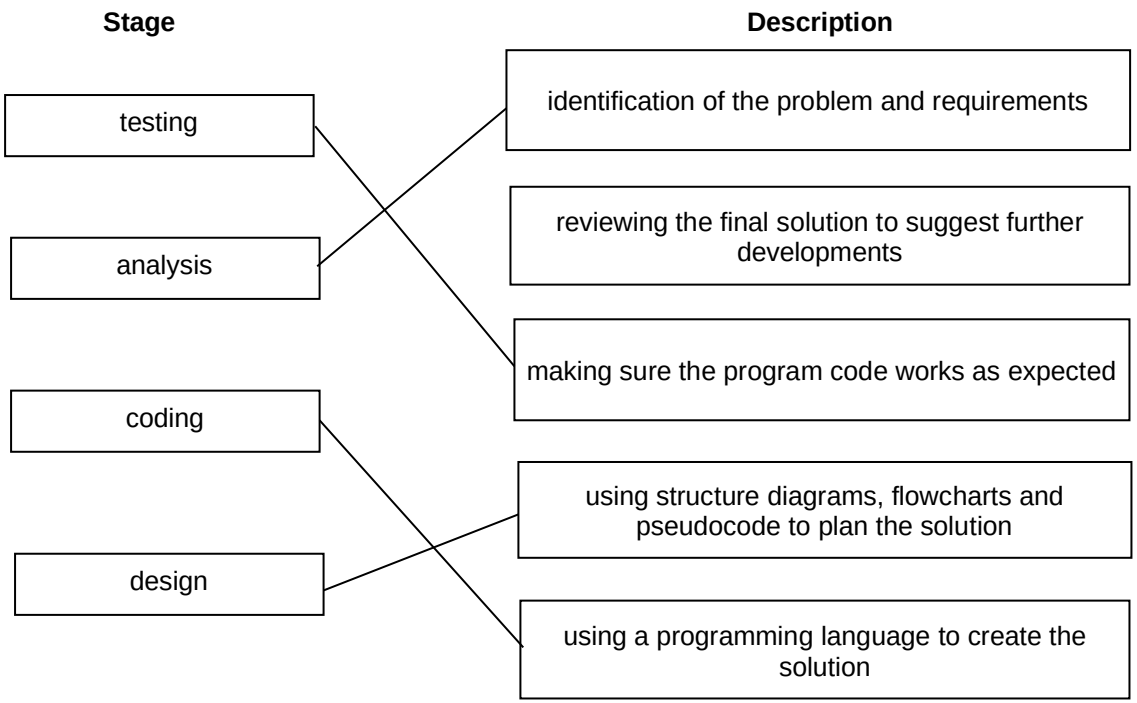
Cambridge Assessment International Education

© UCLES 2025

Annotation	Meaning
	Correct awarding seven marks
	Correct awarding eight marks
	Correct awarding nine marks

Cambridge Assessment International Education

© UCLES 2025

Question	Answer	Marks	Guidance
1	C	**1**	
2	B	**1**	
3	**One** mark for each correct line, **max four**  	**4**	

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>4(a)</td><td>To check that a value has been entered.</td><td>1</td><td>Alternative phrasing of the correct answer is acceptable.</td></tr>
<tr><td>4(b)(i)</td><td>Type (check)</td><td>1</td><td></td></tr>
<tr><td>4(b)(ii)</td><td>One mark per mark point, max five
MP1 Condition controlled loop used
MP2 Input of value into declared variable
MP3 Use of selection to test if input is an integer
MP4 Appropriate use of error message
MP5 Correct condition used to exit/terminate loop

For example
<pre>REPEAT
    INPUT Number
    IF MOD(Number, 1) <> 0
      THEN
        OUTPUT "Please try again"
    ENDIF
UNTIL MOD(Number, 1) = 0</pre></td><td>5</td><td>Alternative with use of DIV e.g. IF DIV(Number, 2) <> Number /2 is acceptable</td></tr>
<tr><td>5</td><td>One mark for each correct box, max six

<table>
<tr><th>Test data</th><th>Type of test data</th><th>Purpose of test data</th></tr>
<tr><td>ABC</td><td>Abnormal</td><td>to make sure that the program rejects data that is too short</td></tr>
<tr><td>Password1</td><td>Boundary</td><td>to make sure that the program rejects data that is only just too short</td></tr>
<tr><td>CambridgeInternational</td><td>Normal</td><td>to make sure that the program accepts data that is an appropriate length</td></tr>
</table></td><td>6</td><td>Shaded parts given in question.

The test data are examples only – accept any correct test data.

Accept alternative correct phrasing of the purposes.</td></tr>
</table>

Question	Answer	Marks	Guidance
6(a)	**One** mark per mark point, **max five** • Line 02 / `DECLARE Highest : STRING` should be `DECLARE Highest : INTEGER` • Line 05 / Highest ← `1500` should be Highest ← `0` • Line 09 / `Total ← Total + Count` should be `Total ← Total + Numbers[Count]` • Line 10 / `IF Numbers[Count] > Total` should be `IF Numbers[Count] > Highest` • Line 17 / `OUTPUT "The average is ", Average / 1000` should be `OUTPUT "The average is ", Total / 1000`	5	Allow any low number – discuss at STM.
6(b)	**One** mark per mark point, **max two** MP1 Correct use of ROUND function to round the average ... MP2 … set to 2 decimal places with `OUTPUT` command. **For example** `OUTPUT "The average is ", ROUND(Total / 1000, 2)`	2	Do not penalise missing output message.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th><th>Guidance</th></tr>
<tr><td>6(c)</td><td>One mark per mark point, max four
MP1 Declaration of new variable for smallest value at start of algorithm
MP2 Initialisation of smallest variable to a high number
MP3 New selection statement after input to compare input with current smallest number
MP4 If input is smaller than current smallest variable, it should replace it
MP5 Outside the loop, output the current value of the smallest variable.</td><td>4</td><td>Possible alternative answers.

MP2 Initialisation of smallest variable to a high number / use of a Boolean/flag to show if the low number has been set (yet).

MP3 Check if smallest value flag not set / check if counter = 1

MP4 OR if smallest value flag not set / if counter = 1</td></tr>
<tr><td>7(a)</td><td>One mark per correct column, max five

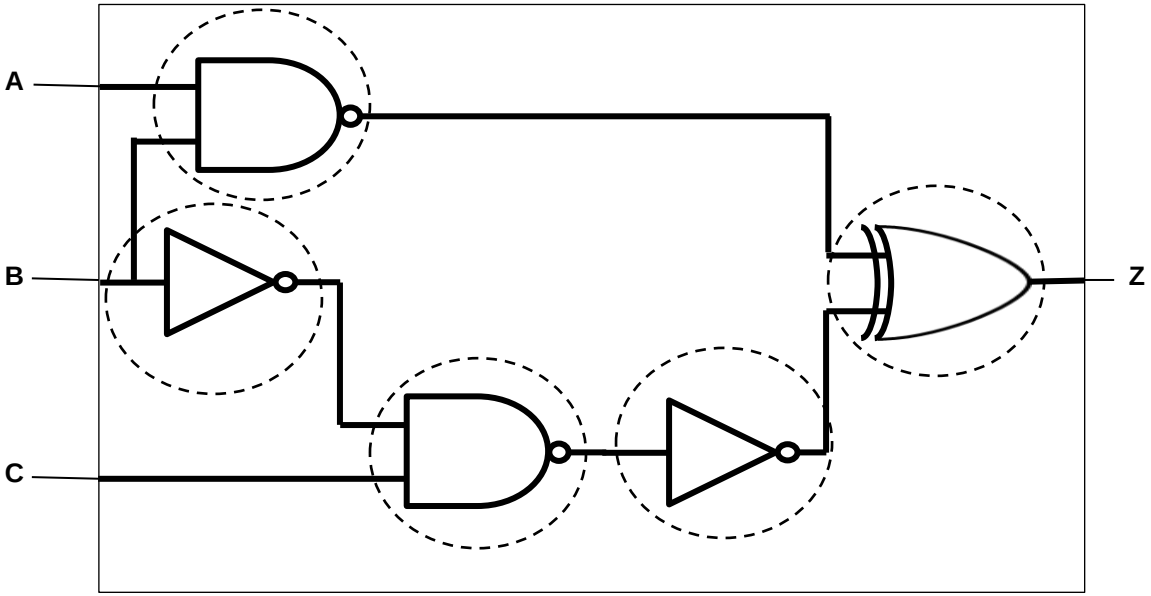
<table>
<tr><th>Answer</th><th>Value1</th><th>Operator</th><th>Value2</th><th>OUTPUT</th></tr>
<tr><td></td><td></td><td></td><td></td><td>Continue?</td></tr>
<tr><td>Y</td><td>7</td><td>S</td><td>9</td><td>−2</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Continue?</td></tr>
<tr><td>Y</td><td>5</td><td>M</td><td>12</td><td>60</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Continue?</td></tr>
<tr><td>Y</td><td>25</td><td>D</td><td>5</td><td>5</td></tr>
<tr><td></td><td></td><td></td><td></td><td>Continue?</td></tr>
<tr><td>N</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td>5</td><td>Ignore case in OUTPUT column, but punctuation must be correct.</td></tr>
<tr><td>7(b)</td><td>A calculator</td><td>1</td><td>Alternative phrasing of the correct answer is acceptable.</td></tr>
</table>

Question	Answer	Marks	Guidance
7(c)(i)	**One** mark per mark point, **max three** MP1 Include two process boxes MP2 … after the `Answer` and `Operator` input boxes MP3 Use of `UCASE` in both boxes to change the input given to upper case. **OR** **One** mark per mark point, **max three** MP1 Change the condition in all the (4) decision boxes MP2 … for the variables `Answer` and `Operator` MP3 Using OR e.g. `OR Answer = 'y'` and `OR Operator = 'a'`	**3**	
7(c)(ii)	**One** mark for example: Only tests for A, S, and M; any other input is assumed to be D (for division) Divide by 0 error (if D and 0 are entered) Wrong data type e.g. character instead of number No data entered (enter without a preceding value).	**1**	Alternative phrasing of the correct answer is acceptable. Ignore case if input values quoted.
8(a)	**One** mark per correct answer, **max two** **Fields:** 6 **Records:** 23	**2**	
8(b)	**One** mark per correct answer, **max two** **Primary key field:** Code **Reason:** It is the only field with unique contents. // It is a unique identifier	**2**	Field must be spelt correctly, but ignore case.

Question	Answer	Marks	Guidance

8(c)	**One** mark per mark point, **max three**
	MP1 Correct data – spelt correctly
	MP2 Correct layout – three columns, with no additional punctuation
	MP3 Correct order
	
	Correct output
	`Valencia Venezuela 1,983,445`
	`Lima Peru 11,206,000`
	`Buenos Aires Argentina 15,490,415`
		3	Allow data presented in tabular form.
			Ignore headings.
			Spelling must be correct, but ignore case.

8(d)	**One** mark per mark point, **max four**
	MP1 At least two correct fields in `SELECT`
	MP2 Remaining two correct fields in `SELECT`
	MP3 Correct `FROM MajorCity`
	MP4 Correct `WHERE Capital = TRUE;`
	
	Correct code:
	`SELECT Code, City, Country, Continent`
	`FROM MajorCity`
	`WHERE Capital = TRUE;`
		4	Ignore case on field names, but spelling must match names in table.
			
			Ignore missing `;` at the end.
			
			Capital is BOOLEAN, but allow "TRUE" or 'TRUE', in case Text is assumed.

Question	Answer	Marks	Guidance
9(a)	**One** mark for each correct gate, with the correct input(s) as shown.  	5	

Question	Answer	Marks	Guidance
9(b)	**Four** marks for eight correct outputs **Three** marks for six or seven correct outputs **Two** marks for four or five correct outputs **One** mark for two or three correct outputs	4	Shaded parts given in question.

A	B	C	Z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Question	Answer	Marks	Guidance
10	Read and understand the question before starting to mark any scripts. Read the whole answer before marking a script. Check if each requirement listed below has been met. Requirements may be met using a suitable built-in function from the programming language used (Python, VB.NET or Java). On the script, add seen if the requirement has been met, NE if a partial attempt, or a cross if no attempt (see marked scripts). Use the tables for AO2 and AO3 below to award a mark in a suitable band using a best fit approach, then add up the total. Marks are available for: • AO2 (maximum 9 marks) • AO3 (maximum 6 marks) **Data structures required** with names as given in the scenario: Arrays or lists `MemberID[]`, `Name[]` Variables `NewID` **Requirements (techniques)** **R1** displays menu and allows choice, then proceeds based on valid choice, (nested iteration, input, output, selection, validation). **R2** checks length of string, checks contents of array for matches, stores approved data (validation, string handling (length), selection, input, storage). **R3** outputs contents of arrays. Program continues until user chooses stop. (output, iteration).	**15**	

Question	Answer	Marks	Guidance

Example 15 mark answer in pseudocode

```
//Array and variable declaration - not required
DECLARE MemberID : ARRAY[1:1000] OF STRING
DECLARE Name : ARRAY[1:1000, 1:2] OF STRING
DECLARE Index, IndexCheck, IndexOut : Integer
DECLARE Answer : INTEGER
DECLARE Code : STRING
DECLARE Used : BOOLEAN
//Initialisation of arrays
FOR Index ← 1 TO 1000
    MemberID[Index] ← ""
    Name[Index, 1] ← ""
    Name[Index, 2] ← ""
NEXT Index
REPEAT
    //Display of menu choices
    OUTPUT "Enter 1 to input a new member, 2 to output member codes and
    first and last names, or 3 to stop"
    //Input menu choice with validation of input
    REPEAT
        INPUT Answer
        IF Answer < 1 OR Answer > 3
          THEN
            OUTPUT "You must input 1, 2 or 3. Please try again"
        ENDIF
    UNTIL Answer >= 1 AND Answer <= 3
    //User chooses to input new member details
    IF Answer = 1
      THEN
        REPEAT
            //Initialisation of flag for previous use of code
            Used ← FALSE
```

Question	Answer	Marks	Guidance

```
//user enters new code, with prompt
OUTPUT "Enter a new six character membership code "
INPUT Code
//Checking code is 6 characters long
IF LENGTH(Code) <> 6
  THEN
    //If code is wrong length, re-entry required
    OUTPUT "The code must contain six characters, please
    try again"
  ELSE
    //If code is correct length, it is checked with
    //with previous codes for uniqueness
    IndexCheck ← 1
    WHILE MemberID[IndexCheck] <> "" AND NOT Used DO
        IF Code = MemberID[IndexCheck]
          THEN
            //If code already used, flag changed and
            //re-entry required
            Used ← TRUE
            OUTPUT "This code has already been used,
            please try again "
          ELSE
            IndexCheck ← IndexCheck + 1
        ENDIF
    ENDWHILE
    //If code correct length and not previously used
    //it is stored, along with first and last name
    IF NOT Used
      THEN
        MemberID[IndexCheck] ← Code
        OUTPUT "Enter your first name "
        INPUT Name[IndexCheck, 1]
        OUTPUT "Enter your last name "
```

Question	Answer	Marks	Guidance

```
                    INPUT Name[IndexCheck, 2]
                ENDIF
            ENDIF
        UNTIL Length(Code) = 6 AND NOT Used
    ENDIF
    //User chooses to output all member details
    IF Answer = 2
      THEN
        IndexOut ← 1
        //All array contents output using iteration
        WHILE MemberID[IndexOut] <> ""
            OUTPUT "Membership code: ", MemberID[IndexOut]
            OUTPUT "First name: ", Name[IndexOut, 1]
            OUTPUT "Last name: ", Name[IndexOut, 2]
            IndexOut ← IndexOut + 1
        ENDWHILE
    ENDIF
UNTIL Answer = 3
```

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 the 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 the 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.*

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 attempts to meet most of the requirements. *Solution contains lines of code that attempt 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.*