



Cambridge O Level

COMPUTER SCIENCE

2210/23

Paper 2 Algorithms, Programming and Logic

October/November 2023

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 2023 series for most Cambridge IGCSE, Cambridge International A and AS Level components, and some Cambridge O Level components.

This document consists of **16** printed pages.

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.

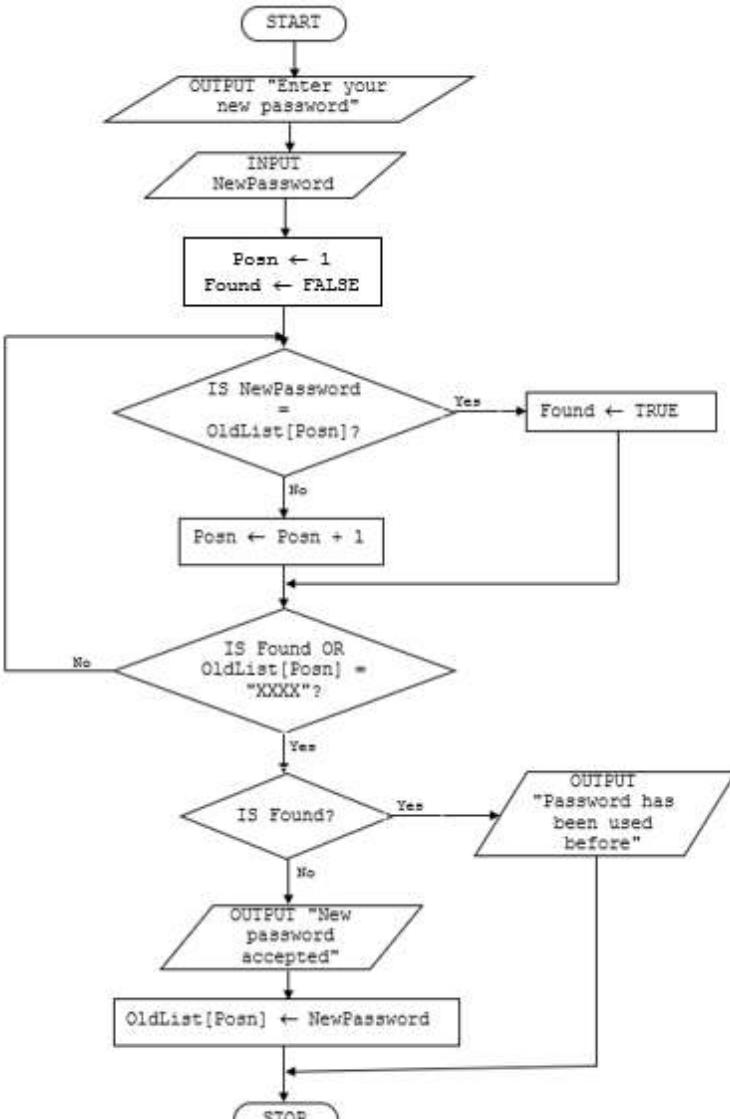
Question	Answer	Marks
1	A	1

Question	Answer	Marks
2	B	1

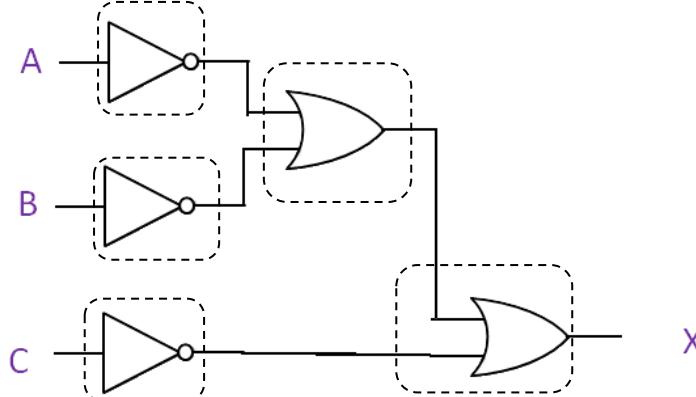
Question	Answer	Marks										
3(a)	<p>One mark for each correct line from description to pseudocode keyword</p> <table> <thead> <tr> <th>Pseudocode description</th> <th>Pseudocode keyword</th> </tr> </thead> <tbody> <tr> <td>stores data in a file</td> <td>OUTPUT</td> </tr> <tr> <td>retrieves data from a file</td> <td>WRITE</td> </tr> <tr> <td>displays data on a screen</td> <td>READ</td> </tr> <tr> <td>enters data from a keyboard</td> <td>OPEN INPUT</td> </tr> </tbody> </table>	Pseudocode description	Pseudocode keyword	stores data in a file	OUTPUT	retrieves data from a file	WRITE	displays data on a screen	READ	enters data from a keyboard	OPEN INPUT	4
Pseudocode description	Pseudocode keyword											
stores data in a file	OUTPUT											
retrieves data from a file	WRITE											
displays data on a screen	READ											
enters data from a keyboard	OPEN INPUT											
3(b)	<p>One mark for each point (max two)</p> <ul style="list-style-type: none"> • data is stored permanently • data can be moved to another computer • another copy of data can be made and stored//accessed elsewhere // backup copy 	2										

Question	Answer	Marks
4(a)	One mark for each point <ul style="list-style-type: none"> • type check • range check 	2
4(b)	One mark for each point (max five) <ul style="list-style-type: none"> • use of loop for check • checking for whole number • checking for number greater than or equal to one • ... and less than or equal to six • Appropriate error/reinput message • ability to reinput value <p>Example:</p> <pre>WHILE Seats < 1 OR Seats > 6 OR Seats <> ROUND(Seats, 0) DO OUTPUT "Please enter a valid number of seats " INPUT Seats ENDWHILE</pre>	5
4(c)	One mark for correct test data, one mark for corresponding reason	2

Question	Answer	Marks
5(a)	One mark for each error identified and correction given <ul style="list-style-type: none"> • Line 06 Password should be NewPassword • Line 11 AND should be OR • Line 16 INPUT should be OUTPUT 	3

Question	Answer	Marks
5(b)	<p>Max six marks from:</p> <p>Max four from:</p> <ul style="list-style-type: none"> one mark for data entry with message one mark for initialisation one mark for checking list // decision box comparing input with array one mark for updating // updating the two variables position and found <p>one mark for loop control // second decision box</p> <p>one mark for setting new password to position in list</p> <p>one mark for outputs // two outputs</p> <p>Two marks:</p> <ul style="list-style-type: none"> one mark for correct use of flow chart symbols one mark for correct use arrows and labels  <pre> graph TD START([START]) --> INPUT[/INPUT "NewPassword"] INPUT --> INIT[Posn ← 1 Found ← FALSE] INIT --> LOOP{IS NewPassword = OldList[Posn] ?} LOOP -- No --> INCR[Posn ← Posn + 1] INCR --> LOOP LOOP -- Yes --> FOUND[Found ← TRUE] FOUND --> CHECK{IS Found OR OldList[Posn] = "XXXX" ?} CHECK -- No --> ACCEPT[/OUTPUT "New password accepted"] ACCEPT --> OLD[OldList[Posn] ← NewPassword] OLD --> STOP([STOP]) CHECK -- Yes --> FOUND2{IS Found ?} FOUND2 -- Yes --> OUTPUT[/OUTPUT "Password has been used before"] FOUND2 -- No --> ACCEPT </pre>	6

Question	Answer	Marks															
6(a)	<p>One mark for correct gate and one mark for correct truth table</p> <p>AND</p> <table border="1" data-bbox="327 323 624 732"> <thead> <tr> <th data-bbox="327 323 444 382">A</th><th data-bbox="444 323 561 382">B</th><th data-bbox="561 323 624 382">X</th></tr> </thead> <tbody> <tr> <td data-bbox="327 382 444 477">0</td><td data-bbox="444 382 561 477">0</td><td data-bbox="561 382 624 477">0</td></tr> <tr> <td data-bbox="327 477 444 573">0</td><td data-bbox="444 477 561 573">1</td><td data-bbox="561 477 624 573">0</td></tr> <tr> <td data-bbox="327 573 444 668">1</td><td data-bbox="444 573 561 668">0</td><td data-bbox="561 573 624 668">0</td></tr> <tr> <td data-bbox="327 668 444 732">1</td><td data-bbox="444 668 561 732">1</td><td data-bbox="561 668 624 732">1</td></tr> </tbody> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	2
A	B	X															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
6(b)	<p>One mark for correct gate and one mark for correct truth table</p> <p>XOR // EOR</p> <table border="1" data-bbox="327 895 624 1303"> <thead> <tr> <th data-bbox="327 895 444 954">A</th><th data-bbox="444 895 561 954">B</th><th data-bbox="561 895 624 954">X</th></tr> </thead> <tbody> <tr> <td data-bbox="327 954 444 1049">0</td><td data-bbox="444 954 561 1049">0</td><td data-bbox="561 954 624 1049">0</td></tr> <tr> <td data-bbox="327 1049 444 1144">0</td><td data-bbox="444 1049 561 1144">1</td><td data-bbox="561 1049 624 1144">1</td></tr> <tr> <td data-bbox="327 1144 444 1240">1</td><td data-bbox="444 1144 561 1240">0</td><td data-bbox="561 1144 624 1240">1</td></tr> <tr> <td data-bbox="327 1240 444 1303">1</td><td data-bbox="444 1240 561 1303">1</td><td data-bbox="561 1240 624 1303">0</td></tr> </tbody> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0	2
A	B	X															
0	0	0															
0	1	1															
1	0	1															
1	1	0															

Question	Answer	Marks															
6(c)	<p>One mark for correct gate and one mark for correct truth table</p> <p>NOR</p> <table border="1" data-bbox="339 357 624 770"> <thead> <tr> <th data-bbox="339 357 428 417">A</th> <th data-bbox="428 357 518 417">B</th> <th data-bbox="518 357 624 417">X</th> </tr> </thead> <tbody> <tr> <td data-bbox="339 452 428 512">0</td> <td data-bbox="428 452 518 512">0</td> <td data-bbox="518 452 624 512">1</td> </tr> <tr> <td data-bbox="339 547 428 608">0</td> <td data-bbox="428 547 518 608">1</td> <td data-bbox="518 547 624 608">0</td> </tr> <tr> <td data-bbox="339 643 428 703">1</td> <td data-bbox="428 643 518 703">0</td> <td data-bbox="518 643 624 703">0</td> </tr> <tr> <td data-bbox="339 738 428 798">1</td> <td data-bbox="428 738 518 798">1</td> <td data-bbox="518 738 624 798">0</td> </tr> </tbody> </table>	A	B	X	0	0	1	0	1	0	1	0	0	1	1	0	2
A	B	X															
0	0	1															
0	1	0															
1	0	0															
1	1	0															
6(d)	<p>One mark for each correct gate, with the correct input(s) as shown.</p> 	5															

Question	Answer	Marks
7	<p>one mark for first description one mark for matching difference max four</p> <ul style="list-style-type: none">• local variables - scope is a defined block of code/subroutine/procedure/function• global variables – scope is the whole program• local variables - value cannot be changed elsewhere in the program• global variables – value can be changed anywhere in the program	4

Question	Answer					Marks																																																																																						
8(a)	<table border="1"> <thead> <tr> <th data-bbox="339 215 496 266">Accept</th><th data-bbox="496 215 653 266">Reject</th><th data-bbox="653 215 810 266">PartOK</th><th data-bbox="810 215 968 266">Error</th><th data-bbox="968 215 1534 266">OUTPUT</th><th data-bbox="1534 215 1545 266"></th></tr> </thead> <tbody> <tr> <td data-bbox="339 266 496 346">0</td><td data-bbox="496 266 653 346">0</td><td data-bbox="653 266 810 346"></td><td data-bbox="810 266 968 346"></td><td data-bbox="968 266 1534 346"></td><td data-bbox="1534 266 1545 346"></td></tr> <tr> <td data-bbox="339 346 496 425">1</td><td data-bbox="496 346 653 425"></td><td data-bbox="653 346 810 425">Y</td><td data-bbox="810 346 968 425"></td><td data-bbox="968 346 1534 425"></td><td data-bbox="1534 346 1545 425"></td></tr> <tr> <td data-bbox="339 425 496 504">2</td><td data-bbox="496 425 653 504"></td><td data-bbox="653 425 810 504">Y</td><td data-bbox="810 425 968 504"></td><td data-bbox="968 425 1534 504"></td><td data-bbox="1534 425 1545 504"></td></tr> <tr> <td data-bbox="339 504 496 584">3</td><td data-bbox="496 504 653 584"></td><td data-bbox="653 504 810 584">Y</td><td data-bbox="810 504 968 584"></td><td data-bbox="968 504 1534 584"></td><td data-bbox="1534 504 1545 584"></td></tr> <tr> <td data-bbox="339 584 496 663"></td><td data-bbox="496 584 653 663">1</td><td data-bbox="653 584 810 663">N</td><td data-bbox="810 584 968 663"></td><td data-bbox="968 584 1534 663"></td><td data-bbox="1534 584 1545 663"></td></tr> <tr> <td data-bbox="339 663 496 743">4</td><td data-bbox="496 663 653 743"></td><td data-bbox="653 663 810 743">Y</td><td data-bbox="810 663 968 743"></td><td data-bbox="968 663 1534 743"></td><td data-bbox="1534 663 1545 743"></td></tr> <tr> <td data-bbox="339 743 496 822">5</td><td data-bbox="496 743 653 822"></td><td data-bbox="653 743 810 822">Y</td><td data-bbox="810 743 968 822"></td><td data-bbox="968 743 1534 822"></td><td data-bbox="1534 743 1545 822"></td></tr> <tr> <td data-bbox="339 822 496 901">6</td><td data-bbox="496 822 653 901"></td><td data-bbox="653 822 810 901">Y</td><td data-bbox="810 822 968 901"></td><td data-bbox="968 822 1534 901"></td><td data-bbox="1534 822 1545 901"></td></tr> <tr> <td data-bbox="339 901 496 981">7</td><td data-bbox="496 901 653 981"></td><td data-bbox="653 901 810 981">Y</td><td data-bbox="810 901 968 981"></td><td data-bbox="968 901 1534 981"></td><td data-bbox="1534 901 1545 981"></td></tr> <tr> <td data-bbox="339 981 496 1060"></td><td data-bbox="496 981 653 1060">2</td><td data-bbox="653 981 810 1060">N</td><td data-bbox="810 981 968 1060"></td><td data-bbox="968 981 1534 1060"></td><td data-bbox="1534 981 1545 1060"></td></tr> <tr> <td data-bbox="339 1060 496 1140">8</td><td data-bbox="496 1060 653 1140"></td><td data-bbox="653 1060 810 1140">Y</td><td data-bbox="810 1060 968 1140"></td><td data-bbox="968 1060 1534 1140"></td><td data-bbox="1534 1060 1545 1140"></td></tr> <tr> <td data-bbox="339 1140 496 1219">9</td><td data-bbox="496 1140 653 1219"></td><td data-bbox="653 1140 810 1219">Y</td><td data-bbox="810 1140 968 1219"></td><td data-bbox="968 1140 1534 1219"></td><td data-bbox="1534 1140 1545 1219"></td></tr> <tr> <td data-bbox="339 1219 496 1298">10</td><td data-bbox="496 1219 653 1298"></td><td data-bbox="653 1219 810 1298">Y</td><td data-bbox="810 1219 968 1298">20</td><td data-bbox="968 1219 1534 1298"></td><td data-bbox="1534 1219 1545 1298"></td></tr> <tr> <td data-bbox="339 1298 496 1378"></td><td data-bbox="496 1298 653 1378"></td><td data-bbox="653 1298 810 1378"></td><td data-bbox="810 1298 968 1378"></td><td data-bbox="968 1298 1534 1378">Too many rejected 20% error</td><td data-bbox="1534 1298 1545 1378"></td></tr> </tbody> </table>	Accept	Reject	PartOK	Error	OUTPUT		0	0					1		Y				2		Y				3		Y					1	N				4		Y				5		Y				6		Y				7		Y					2	N				8		Y				9		Y				10		Y	20							Too many rejected 20% error		5
Accept	Reject	PartOK	Error	OUTPUT																																																																																								
0	0																																																																																											
1		Y																																																																																										
2		Y																																																																																										
3		Y																																																																																										
	1	N																																																																																										
4		Y																																																																																										
5		Y																																																																																										
6		Y																																																																																										
7		Y																																																																																										
	2	N																																																																																										
8		Y																																																																																										
9		Y																																																																																										
10		Y	20																																																																																									
				Too many rejected 20% error																																																																																								
One mark for each column																																																																																												

Question	Answer	Marks
8(b)	<p>One mark for each point max three</p> <ul style="list-style-type: none"> • after the Input box // before the first decision box • insert a process box • to convert the input to upper case <p>OR</p> <ul style="list-style-type: none"> • change the first decision / add another decision box • to accept 'y' as well • by adding OR PartOK = 'y' 	3

Question	Answer	Marks				
9(a)	Records: 14 Fields: 5	2				
9(b)(i)	Species/Description	1				
9(b)(ii)	Long names that could be easily misspelt // species or description could be duplicated	1				
9(b)(iii)	Easy to validate // always unique	1				
9(c)	<p>One mark for each correct row or column</p> <table> <tr> <td>True silver</td> <td>white laced top half and black lower half</td> </tr> <tr> <td>Brown eared</td> <td>brown with ear tufts</td> </tr> </table>	True silver	white laced top half and black lower half	Brown eared	brown with ear tufts	2
True silver	white laced top half and black lower half					
Brown eared	brown with ear tufts					
9(d)	<p>One mark for each correct addition</p> <pre>SELECT Species FROM PheasantList WHERE Breeding or WHERE Young = 0 AND Young = 0; AND Breeding;</pre>	4				

Question	Answer	Marks
10	<ul style="list-style-type: none"> AO2 (maximum 9 marks) AO3 (maximum 6 marks) <p>Data Structures required names shown underlined must be used as given in the scenario 2D Array or list <u>Temperatures</u> Variables <u>MaxDay</u>, <u>MinDay</u>, <u>AvDay</u>, <u>MaxWeek</u>, <u>MinWeek</u>, <u>AvWeek</u></p> <p>Requirements (techniques)</p> <p>R1 Find maximum and minimum temperatures for each day and calculates the average daily temperature (searching, totalling)</p> <p>R2 Find maximum and minimum temperatures for week and calculates the average weekly temperature (nested searching, totalling)</p> <p>R3 outputs for each day name, the rounded values for maximum temperature, minimum temperatures and average temperature. Outputs for the week the rounded values for maximum temperature, minimum temperatures and average temperature (output with appropriate messages and rounded values)</p> <p>Example 15-mark answer in pseudocode:</p> <pre> // meaningful identifier names and appropriate data structures to store the data required DECLARE DayCounter, HourCounter : INTEGER DECLARE AvDay, AvWeek, MaxDay, MinDay, MaxWeek, MinWeek : REAL DECLARE DayTotal, WeekTotal : REAL DECLARE Day : STRING CONSTANT Hours ← 24 CONSTANT Days ← 7 </pre>	15

Question	Answer	Marks
10	<pre> MaxWeek ← -1000// initialise max and min temperatures and total for the week MinWeek ← 1000 WeekTotal ← 0 FOR DayCounter ← 0 TO Days - 1 MaxDay ← -1000// initialise max and min temperatures and total for each day MinDay ← 1000 DayTotal ← 0 FOR HourCounter ← 0 TO Hours - 1 DayTotal ← DayTotal + Temperatures(HourCounter, DayCounter) // update total maximum and minimum IF Temperatures(HourCounter, DayCounter) > MaxDay THEN MaxDay ← Temperatures(HourCounter, DayCounter) ENDIF IF Temperatures(HourCounter, DayCounter) < MinDay THEN MinDay ← Temperatures(HourCounter, DayCounter) ENDIF NEXT HourCounter CASE OF DayCounter // select message for day 0 : Day ← "Monday" 1 : Day ← "Tuesday" 2 : Day ← "Wednesday" 3 : Day ← "Thursday" 4 : Day ← "Friday" 5 : Day ← "Saturday" 6 : Day ← "Sunday" ENDCASE DayAverage ← DayTotal / Hours // output results for day OUTPUT Day // Results from a day OUTPUT "Maximum temperature ", MaxDay OUTPUT "Minimum temperature ", MinDay OUTPUT "Average temperature ", ROUND(DayAverage,2) </pre>	

Question	Answer	Marks
10	<pre> IF MaxDay > MaxWeek // update total maximum and minimum THEN MaxWeek ← MaxDay ENDIF IF MinDay > MinWeek THEN MinWeek ← MinDay ENDIF WeekTotal ← WeekTotal + DayTotal // update total for week NEXT DayCounter WeekAverage ← WeekTotal / Days OUTPUT "Maximum temperature for week ", MaxWeek// output results for week OUTPUT "Minimum temperature for week ", MinWeek OUTPUT "Average temperature for Week ", ROUND(WeekAverage,2) </pre>	

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. <i>Any use of selection, iteration, counting, totalling, input and output.</i>	Some programming techniques used are appropriate to the problem. <i>More than one technique seen applied to the scenario, check the list of techniques needed.</i>	The range of programming techniques used is appropriate to the problem. <i>All criteria stated for the scenario have been covered by the use of appropriate programming techniques, check list of techniques needed.</i>
	Some data has been stored but not appropriately. <i>Any use of variables or arrays or other language dependent data structures e.g. Python lists.</i>	Some of the data structures chosen are appropriate and store some of the data required. <i>More than one data structure used to store data required by the scenario.</i>	The data structures chosen are appropriate and store all the data required. <i>The data structures used store all the data required by the scenario.</i>

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.	<p>Program seen without relevant comments.</p> <p><i>Some of the data structures used have meaningful names.</i></p>	<p>Program seen with some relevant comment(s).</p> <p>The majority of identifiers used are appropriately named.</p> <p><i>Most of the data structures used have meaningful names.</i></p>	<p>The program has been fully commented.</p> <p>Suitable identifiers with names meaningful to their purpose have been used throughout.</p> <p><i>All of the data structures used have meaningful names.</i></p>
	<p>The solution is illogical.</p> <p>The solution is inaccurate in many places.</p> <p><i>Solution contains few lines of code with errors that attempt to perform a task given in the scenario.</i></p>	<p>The solution contains parts that may be illogical.</p> <p>The solution contains parts that are inaccurate.</p> <p><i>Solution contains lines of code with some errors that logically perform tasks given in the scenario. Ignore minor syntax errors.</i></p>	<p>The program is in a logical order.</p> <p>The solution is accurate.</p> <p><i>Solution logically performs all the tasks given in the scenario. Ignore minor syntax errors.</i></p>
	<p>The solution attempts at least one of the requirements.</p> <p><i>Solution contains lines of code that attempt at least one task given in the scenario.</i></p>	<p>The solution attempts to meet most of the requirements.</p> <p><i>Solution contains lines of code that perform most tasks given in the scenario.</i></p>	<p>The solution meets all the requirements given in the question.</p> <p><i>Solution performs all the tasks given in the scenario.</i></p>