Cambridge International AS & A Level

COMPUTER SCIENCE **9618/43**

Paper 4 Practical **May/June 2021**

MARK SCHEME

Maximum Mark: 75

Published

This document consists of **30** printed pages.

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 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 mark per bullet point
	• Declaring record/class with name node…
	• …declaring data and next node (both as Integers)
	
	Example code:
	
	Visual Basic
	<pre>Structure node
	    Dim Data As Integer
	    Dim nextNode As Integer
	End Structure</pre>
	Python
	<pre>class node:
	    def __init__(self, theData, nextNodeNumber):
	        self. Data = theData
	        self.nextNode = nextNodeNumber</pre>
	Java
	<pre>class node{
	  private Integer Data;
	  private Integer nextNode;
	  public node(Integer dataP, Integer nextNodeP){
	    this.Data = dataP;
	    this.nextNode = nextNodeP;
	  }
	}</pre>
		2

Question	Answer	Marks
1(b)	1 mark per bullet point • Declaring array named `linkedList` with data type `node` • Assigning all nodes correctly as record/object nodes … • …with correct values stored • declaring `startPointer` as 0, `emptyList` as 5 Example code: **Visual Basic** <code>Dim linkedList(9) As node</code> <code>linkedList(0).data = 1</code> <code>linkedList(0).nextNode = 1</code> <code>linkedList(1).data = 5</code> <code>linkedList(1).nextNode = 4</code> <code>linkedList(2).data = 6</code> <code>linkedList(2).nextNode = 7</code> <code>linkedList(3).data = 7</code> <code>linkedList(3).nextNode = -1</code> <code>linkedList(4).data = 2</code> <code>linkedList(4).nextNode = 2</code> <code>linkedList(5).data = 0</code> <code>linkedList(5).nextNode = 6</code> <code>linkedList(6).data = 0</code> <code>linkedList(6).nextNode = 8</code> <code>linkedList(7).data = 56</code> <code>linkedList(7).nextNode = 3</code> <code>linkedList(8).data = 0</code> <code>linkedList(8).nextNode = 9</code> <code>linkedList(9).data = 0</code> <code>linkedList(9).nextNode = -1</code> <code>Dim startPointer As Integer = 0</code> <code>Dim emptyList As Integer = 5</code>	4

Question	Answer	Marks

1(b)	**Python**
	``
	linkedList = [node(1,1),node(5,4),node(6,7),node(7,-1),node(2,2),node(0,6),
	             node(0,8),node(56,3),node(0,9),node(0,-1)]
	startPointer = 0
	emptyList = 5
	``
	
	Java
	``
	public static void main(String[] args){
	    node[] linkedList = new node[10];
	    linkedList[0] = new node(1,1);
	    linkedList[1] = new node(5, 4);
	    linkedList[2] = new node(6, 7);
	    linkedList[3] = new node(7,-1);
	    linkedList[4] = new node(2,2);
	    linkedList[5] = new node(0,6);
	    linkedList[6] = new node(0,8);
	    linkedList[7] = new node(56, 3);
	    linkedList[8] = new node(0,9);
	    linkedList[9] = new node(0,-1);
	    Integer startPointer = 0;
	    Integer emptyList = 5;
	}
	``	

Question	Answer	Marks

1(c)(i)	1 mark per bullet point
	• Procedure `outputNodes` …
	• …taking linked list and start pointer as parameters
	• Looping until `nextNode/pointer` is –1
	• Outputting the node data in the correct order, i.e. following pointers
	• Updating pointer to current node's `nextNode`
	• Using the correct record/class field/properties throughout
	
	Example code:
	
	Visual Basic
	<pre>Sub outputNodes(ByRef linkedList, ByVal currentPointer)
	    While (currentPointer <> -1)
	        Console.WriteLine(linkedList(currentPointer).data)
	        currentPointer = linkedList(currentPointer).nextNode
	    End While
	End Sub</pre>
	Python
	<pre>def outputNodes(linkedList, currentPointer):
	    while(currentPointer != -1):
	        print(str(linkedList[currentPointer].data))
	        currentPointer = linkedList[currentPointer].nextNode</pre>
	Java
	<pre>public static void outputNodes(node[] linkedList, Integer currentPointer){
	  while(currentPointer != -1){
	    System.out.println(linkedList[currentPointer].data);
	    currentPointer = linkedList[currentPointer].nextNode;
	  }
	}</pre>
		6

Question	Answer	Marks
1(c)(ii)	Screenshot showing: 1 5 2 6 56 7	**1**
1(d)(i)	1 mark per bullet point to **max 7** • Function taking list and both pointers as parameters • Taking (integer) data as input • Checking if list is full … • … and returning False • Insert the input data to the empty list node's data • Following pointers to find last node in Linked List … • …and updating last node's pointer to empty list/location where new node is added • Updating empty list to it's first elements pointer • Returning true when added successfully Example code: **Visual Basic** ```Function addNode(ByRef linkedList() As node, ByVal currentPointer As Integer, ByRef emptyList As Integer)
Console.WriteLine("Enter the data to add")
Dim dataToAdd As Integer = Console.ReadLine()
Dim previousPointer As Integer = 0
Dim newNode As node
If emptyList < 0 Or emptyList > 9 Then
Return False
Else
newNode.data = dataToAdd
newNode.nextNode = -1```	**7**

Question	Answer	Marks
| 1(d)(i) | ```
            linkedList(emptyList) = newNode
            previousPointer = 0
            While (currentPointer <> -1)
                previousPointer = currentPointer
                currentPointer = linkedList(currentPointer).nextNode
            End While
            Dim valueToWrite As Integer = emptyList
            linkedList(previousPointer).nextNode = valueToWrite
            emptyList = linkedList(emptyList).nextNode

            Return True

        End If

    End Function
```<br><br>**Python**<br>```
def addNode(linkedList, currentPointer, emptyList):
    dataToAdd = input("Enter the data to add")

    if emptyList <0 or emptyList > 9:
        return False
    else:
        newNode = node(int(dataToAdd), -1)
        linkedList[emptyList] = (newNode)

        previousPointer = 0
        while(currentPointer != -1):
            previousPointer = currentPointer
            currentPointer = linkedList[currentPointer].nextNode
        linkedList[previousPointer].nextNode = emptyList
        emptyList = linkedList[emptyList].nextNode

        return True
``` | |

Question	Answer	Marks

1(d)(i)	**Java**
	``java
	public static Boolean addNode(node[] linkedList, Integer currentPointer,
	                            Integer emptyList){
	    Integer dataToAdd;
	    Integer previousPointer;
	    node newNode;
	    Scanner in = new Scanner(System.in);
	    System.out.println("Enter the data to add");
	    dataToAdd = in.nextInt();
	    if(emptyList < 0 || emptyList > 9){
	        return false;
	    }else{
	        newNode = new node(dataToAdd, -1);
	        linkedList[emptyList] = newNode;
	        previousPointer = 0;
	        while(currentPointer != -1){
	            previousPointer = currentPointer;
	            currentPointer = linkedList[currentPointer].nextNode;
	
	        }
	        linkedList[previousPointer].nextNode = emptyList;
	        emptyList = linkedList[emptyList].nextNode;
	        return true;
	    }
	}
	``	

Question	Answer	Marks

1(d)(ii)	1 mark per bullet point
	• Call `addNode()` with list, start and empty pointers and store/check return value …
	• …output appropriate message if True returned and if False returned
	• Calling `outputNodes()` with list and start pointer before and after `addNode()`
	
	Example code:
	
	Visual Basic
	<pre>Sub Main()
	  Dim linkedList(10) As node
	  linkedList(0).data = 1
	  linkedList(0).nextNode = 1
	  linkedList(1).data = 5
	  linkedList(1).nextNode = 4
	  linkedList(2).data = 6
	  linkedList(2).nextNode = 7
	  linkedList(3).data = 7
	  linkedList(3).nextNode = -1
	  linkedList(4).data = 2
	  linkedList(4).nextNode = 2
	  linkedList(5).data = -1
	  linkedList(5).nextNode = 6
	  linkedList(6).data = -1
	  linkedList(6).nextNode = 7
	  linkedList(7).data = 56
	  linkedList(7).nextNode = 3
	  linkedList(8).data = -1
	  linkedList(8).nextNode = 9
	  linkedList(9).data = -1
	  linkedList(9).nextNode = -1
	  Dim startPointer As Integer = 0
	  Dim emptyList As Integer = 5
	  outputNodes(linkedList, startPointer)
	  Dim returnValue As Boolean
	  returnValue = addNode(linkedList, startPointer,
	  emptyList)</pre>
		3

Question	Answer	Marks
1(d)(ii)	<pre> If returnValue = True Then
        Console.WriteLine("Item successfully added")
    Else
        Console.WriteLine("Item not added, list full")
    End If
    outputNodes(linkedList, startPointer)
    Console.ReadLine()
End Sub</pre>**Python**<pre>linkedList = [node(1,1),node(5,4),node(6,7),node(7,-1),node(2,2),node(-1,6),
             node(-1,7),node(56,3),node(-1,9),node(-1,-1)]
startPointer = 0
emptyList = 5
outputNodes(linkedList, startPointer)
returnValue = addNode(linkedList, startPointer, emptyList)
if returnValue == True:
    print("Item successfully added")
else:
    print("Item not added, list full")
outputNodes(linkedList, startPointer)</pre>**Java**<pre>public static void main(String[] args){

    node[] linkedList = new node[10];
    linkedList[0] = new node(1,1);
    linkedList[1] = new node(5, 4);
    linkedList[2] = new node(6, 7);
    linkedList[3] = new node(7,-1);
    linkedList[4] = new node(2,2);
    linkedList[5] = new node(-1,6);
    linkedList[6] = new node(-1,7);
    linkedList[7] = new node(56, 3);
    linkedList[8] = new node(-1,9);</pre> | |

Question	Answer	Marks

1(d)(ii)	```java
	        linkedList[9] = new node(-1,-1);
	        Integer startPointer = 0;
	        Integer emptyList = 5;
	        outputNodes(linkedList, startPointer);
	        Boolean returnValue;
	        returnValue = addNode(linkedList, startPointer, emptyList);
	        if (returnValue == true){
	            System.out.println("Item successfully added");
	        }else{
	            System.out.println("Item not added, list full");
	        }
	        outputNodes(linkedList, startPointer);
	}
	``	

1(d)(iii)	1 mark for screenshot showing : • Linked list output • 5 input • Message saying Successfully added or equivalent • Linked list output with 5 at the end. Example: 1 5 2 6 56 7 5 (being input) 1 5 2 6 56 7 5	1

Question	Answer	Marks

2(a)	1 mark per bullet point
	• Array with identifier `arrayData`
	• correct 10 data items added
	
	Example code:
	
	Visual Basic
	<pre> Dim arrayData(9) As Integer
	Sub Main()
	  arrayData(0) = 10
	  arrayData(1) = 5
	  arrayData(2) = 6
	  arrayData(3) = 7
	  arrayData(4) = 1
	  arrayData(5) = 12
	  arrayData(6) = 13
	  arrayData(7) = 15
	  arrayData(8) = 21
	  arrayData(9) = 8
	End Sub</pre>
	Python
	<pre>arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]</pre>
	Java
	<pre>int[] arrayData = new int[];
	public static void main(String[] args){
	     arrayData[0] = 10;
	     arrayData[1] = 5;
	     arrayData[2] = 6;
	     arrayData[3] = 7;
	     arrayData[4] = 1;
	     arrayData[5] = 12;
	     arrayData[6] = 13;</pre>
		2

Question	Answer	Marks

2(a)	```
	        arrayData[7] = 15;
	        arrayData[8] = 21;
	        arrayData[9] = 8;
	}
	``	

2(b)(i)	1 mark per bullet point
	• function `linearSearch` with correct identifier
	• …taking integer search value as a parameter
	• Searching 10 times/through all array elements …
	• …comparing each element to search value
	• returning True if found
	• returning False if not found
	
	Example code:
	
	Visual Basic
	<pre>Function linearSearch(ByRef searchValue As Integer)
	  For x = 0 To 9
	    If arrayData(x) = searchValue Then
	      Return True
	    End If
	  Next
	  Return False
	End Function</pre>
		6

Question	Answer	Marks

2(b)(i)	**Python**
	
	```python\ndef linearSearch(searchValue):\n for x in range(0, 10):\n if arrayData[x] == searchValue:\n return True\n return False\n```
	
	Java
	
	```java\npublic static Boolean linearSearch(Integer searchValue){\n for (int x = 0; x < 10; x++){\n if(arrayData[x] == searchValue){\n return true;\n }\n }\n return false;\n}\n```	

Question	Answer	Marks

2(b)(ii)	1 mark per bullet point to max 4
	• Taking value as input…
	• …checking/casting to Integer
	• Calling `linearSearch` and sending input as parameter
	• Storing and checking return value…
	• …outputting appropriate message if found **and** if not found
	
	Example code:
	
	Visual Basic
	<pre>Dim arrayData(10) As Integer
	Sub Main()
	  arrayData(0) = 10
	  arrayData(1) = 5
	  arrayData(2) = 6
	  arrayData(3) = 7
	  arrayData(4) = 1
	  arrayData(5) = 12
	  arrayData(6) = 13
	  arrayData(7) = 15
	  arrayData(8) = 12
	  arrayData(9) = 8
	  Console.WriteLine("Enter a number to search for")
	  Dim searchValue As Integer = Console.ReadLine()
	  Dim returnValue As Boolean = linearSearch(searchValue)
	  If returnValue = True Then
	      Console.WriteLine("Found it")
	  Else
	      Console.WriteLine("Didn't find it")
	  End If
	End Sub</pre>
		4

Question	Answer	Marks

2(b)(ii)	**Python**
	``python
	arrayData = [10, 5, 6, 7, 1, 12, 13, 15, 21, 8]
	searchValue = int(input("Enter the number to search for"))
	returnValue = linearSearch(searchValue)
	if returnValue == True:
	    print("It was found")
	else:
	    print("It was not found")
	``
	
	Java
	``java
	Integer[] arrayData = new Integer[10];
	public static void main(String[] args){
	
	    arrayData[0] = 10;
	    arrayData[1] = 5;
	    arrayData[2] = 6;
	    arrayData[3] = 7;
	    arrayData[4] = 1;
	    arrayData[5] = 12;
	    arrayData[6] = 13;
	    arrayData[7] = 15;
	    arrayData[8] = 12;
	    arrayData[9] = 8;
	    System.out.println("Enter the number to search for");
	    Integer searchValue;
	    Scanner in = new Scanner(System.in);
	    searchValue = in.nextInt();
	    Boolean returnValue;
	    returnValue = linearSearch(searchValue);
	    if (returnValue == true){
	        System.out.println("It was found");
	    }else{
	        System.out.println("It was not found");
	    }
	}
	``	

Question	Answer	Marks
2(b)(iii)	1 mark for screenshot showing input and output for number found 1 mark for screenshot showing input and output for number not found	**2**

2(c)	1 mark per bullet point
	• Correct outer loop stop
	• Correct inner loop stop
	• Correct < in the IF
	• Correct `theArray(y + 1)`
	• Correct `temp`
	• Remainder matching pseudocode
	
	Example code:
	
	Visual Basic
	<pre>Sub bubbleSort()
	  Dim temp As Integer = 0
	  For x = 0 To 9
	    For y = 0 To 8
	      If theArray(y) < theArray(y + 1) Then
	        temp = theArray(y)
	        theArray(y) = theArray(y + 1)
	        theArray(y + 1) = temp
	      End If
	    Next
	  Next
	End Sub</pre>
		6

Question	Answer	Marks

2(c)	**Python**
	``python
	def bubbleSort():
	    for x in range (0, 10):
	        for y in range(0, 9):
	            if theArray[y] < theArray[y + 1]:
	                temp = theArray[y]
	                theArray[y] = theArray[y + 1]
	                theArray[y + 1] = temp
	``
	
	Java
	``java
	public static void bubbleSort(){
	    int temp;
	    for (int x = 0; x < 10; x++){
	        for (int y = 0; y < 9; y++){
	            if(theArray[y] < theArray[y+1]){
	                temp = theArray[y];
	                theArray[y] = theArray[y+1];
	                theArray[y+1] = temp;
	            }
	        }
	    }
	}
	``	

Question	Answer	Marks

3(a)	1 mark per bullet point
	• Class named `treasureChest` and end
	• Question declared as string as a class attribute
	• Answer declared as integer as a class attribute
	• Points declared as integer as a class attribute
	• All 3 attributes are private
	
	Example code:
	
	Visual Basic
	<pre>Class treasureChest
	  Private question As String
	  Private answer As Integer
	  Private points As Integer
	
	  Sub New(questionP, answerP, pointsP)
	    question = questionP
	    answer = answerP
	    points = pointsP
	  End Sub
	End Class</pre>**Python**
	<pre>class treasureChest:
	    #Private question : String
	    #Private answer : Integer
	    #Private points : Integer
	
	    def __init__(self, questionP, answerP, pointsP):
	        self.__question = questionP
	        self.__answer = answerP
	        self.__points = points</pre>
		5

Question	Answer	Marks

3(a)	**Java**
	``java
	import java.util.Scanner;
	
	class treasureChest{
	      private String question;
	      private Integer answer;
	      private Integer points;
	
	      public treasureChest(String questionP, Integer answerP, Integer pointsP){
	            question = questionP;
	            answer = answerP;
	            points = pointsP;
	      }
	}
	``	

3(b)	1 mark per bullet point to **max 8** • procedure declared as readData • declare array arrayTreasure with 4 elements type treasureChest • opening correct file for read • looping until EOF/5 questions … • …reading in and storing each group of 3 lines appropriately • creating object of type treasureChest … • …with question, answer and points from file as parameters • ..adding to next array element/appending • … repeatedly for all 5 questions in correct order • Use of appropriate exception handler… • …appropriate output if file not found • Closing correct file	**8**

Question	Answer	Marks

3(b)	Example code:
	
	Visual Basic
	`Sub readData()`
	`  Dim arrayTreasure(4) as treasureChest`
	`  Dim filename As String = "treasureChestData.txt"`
	`  Try`
	`    Dim fileReader As New System.IO.StreamReader(filename)`
	`    Dim question As String`
	`    Dim answer, points As Integer`
	`    Dim numberQuestions as Integer = 0`
	`    While fileReader.Peek <> -1`
	`      question = fileReader.ReadLine()`
	`      answer = fileReader.ReadLine()`
	`      points = fileReader.ReadLine()`
	`       arrayTreasure(numberQuestions) = New treasureChest(question, answer, points)`
	`      numberQuestions += 1`
	`    End While`
	`    fileReader.Close()`
	`  Catch ex As Exception`
	`    Console.WriteLine("Invalid file")`
	`  End Try`
	`End Sub`
	
	Python
	`# arrayTreasure(5) as treasureChest`
	`def readData():`
	`    filename = "treasureChestData.txt"`
	`    try:`
	`        file= open(filename,"r")`
	`        dataFetched = (file.readline()).strip()`
	`        while(dataFetched != "" ):`
	`            question = dataFetched`
	`            answer = (file.readline()).strip()`	

Question	Answer	Marks
3(b)	(see below)	

```
            points = (file.readline()).strip()
            arrayTreasure.append(treasureChest(question, answer, points))
            dataFetched = (file.readline()).strip()
        file.close()
    except IOError:
        print("Could not find file")
```

Java
```
public static void readData(){
        treasureChest[] arrayTreasure = new treasureChest[5]:
    String filename = "treasureChestData.txt";
    String dataRead;
    String question;
    String answer;
    String points;
     Integer numberQuestions = 0;
    try{
        FileReader f = new FileReader(filename);
        BufferedReader reader = new BufferedReader(f);
        dataRead = reader.readLine();

        while (dataRead != null){
            question = dataRead;
            answer = reader.readLine();
            points = reader.readLine();
            arrayTreasure[numberQuestions] = new treasureChest(question,
                            Integer.parseInt(answer), Integer.parseInt(points));
            numberQuestions++;
            dataRead = reader.readLine();
        }
        reader.close();
    }
```

Question	Answer	Marks

3(b)	<pre> catch(FileNotFoundException ex){
	                System.out.println("No file found");
	        }
	        catch(IOException ex){
	                System.out.println("No file found");
	        }
	}</pre>	

3(c)(i)	1 mark for `getQuestion` returning the value of question
	
	Example code:
	
	Visual Basic
	<pre>Function getQuestion()
	  Return question
	End Function</pre>
	Python
	<pre> def getQuestion(self):
	        return self.__question</pre>
	Java
	<pre>public String getQuestion(){
	     return question;
	}</pre>
		1

Question	Answer	Marks

3(c)(ii)	1 mark per bullet point
	• Function checkAnswer taking in the parameter, returning Boolean
	• Comparing parameter to that object's answer…
	• …returning True if correct and False otherwise
	
	Example code:
	
	Visual Basic
	`Function checkAnswer(answerP)`
	`  If answer = answerP Then`
	`    Return True`
	`  Else`
	`    Return False`
	`  End If`
	`End Function`
	
	Python
	`def checkAnswer(self, answerP):`
	`  if int(self.__answer) == answerP:`
	`    return True`
	`  else:`
	`    return False`
	
	Java
	`public Boolean checkAnswer(Integer answerP){`
	`    if (answer == answerP){`
	`        return true;`
	`    }else{`
	`        return false;`
	`    }`
	`}`
		3

Question	Answer	Marks

3(c)(iii)	1 mark per bullet point
	 • Function `getPoints` taking `attempts` as parameter and returning integer
	 • If `attempts` is 1 returning `points`
	 • If `attempts` is 2 returns `points` DIV 2
	 • If `attempts` is 3 or 4 returns `points` DIV 4
	 • otherwise returns 0
	
	 Example code:
	
	 Visual Basic
	 <pre>Function getPoints(attempts)
	  If attempts = 1 Then
	    Return points
	  ElseIf attempts = 2 Then
	    Return points \ 2
	  ElseIf attempts = 3 Or attempts = 4 Then
	    Return points \ 4
	  Else
	    Return 0
	  End If
	End Function</pre> **Python**
	 <pre>def getPoints(self, attempts):
	        if attempts == 1:
	            return int(self.__points)
	        elif attempts == 2:
	            return int(self.__points) // 2
	        elif attempts == 3 or attempts == 4:
	            return int(self.__points) // 4
	        else:
	            return 0</pre>
		5

Question	Answer	Marks

3(c)(iii)	**Java**
	
	``java
	public Integer getPoints(Integer attempts){
	     if (attempts == 1){
	          return points;
	     }else if(attempts == 2){
	          return Math.round(points/2);
	     }else if(attempts == 3 || attempts == 4){
	          return Math.round(points/4);
	     }else{
	          return 0;
	     }
	}
	``	

3(c)(iv)	1 mark per bullet point to **max 7** • Call the procedure `readData()` • Take the question number as input from user • ..validated between 1 and 5 • Output the question stored at user's input value • Read answer from user • Check the answer input against question's answer • …looping until the answer is correct • Keeping track of the number of attempts using a variable • Using `getPoints()` and sending the number of attempts as a parameter … • …outputting the number of points returned • Using `.getQuestion` and `.checkAnswer` to access question number input by user and answer input by used	**7**

Question	Answer	Marks

3(c)(iv)	Example code:
	
	Visual Basic
	<pre>Sub Main()
	  readData()
	  Console.WriteLine("Pick a treasure chest to open")
	  Dim choice As Integer = Console.ReadLine()
	  Dim result As Boolean
	  Dim answer As Integer
	  Dim attempts As Integer = 0
	  If choice > 0 And choice < 6 Then
	    result = False
	    attempts = 0
	    While result = False
	      Console.WriteLine(arrayTreasure(choice - 1).getQuestion())
	      answer = Console.ReadLine
	      result = arrayTreasure(choice - 1).checkAnswer(answer)
	      attempts = attempts + 1
	    End While
	    Console.WriteLine(arrayTreasure(choice - 1).getPoints(attempts))
	  End If
	End Sub</pre>
	Python
	<pre>readData()
	choice = int(input("Pick a treasure chest to open"))
	if choice > 0 and choice < 6:
	
	    result = False
	    attempts = 0
	    while result == False:
	      answer = int(input(arrayTreasure[choice-1].getQuestion()))
	      result = arrayTreasure[choice-1].checkAnswer(answer)
	      attempts = attempts + 1
	     print(int(arrayTreasure[choice-1].getPoints(attempts)))</pre>	

Question	Answer	Marks

3(c)(iv)	**Java**
	<pre>public static void main(String[] args){
	
	    readData();
	    Scanner scanner = new Scanner(System.in);
	    System.out.println("Pick a treasure chest to open");
	    Integer answer;
	    Integer choice;
	    choice= Integer.parseInt(scanner.nextLine());
	    Integer attempts;
	    if (choice> 0 && choice < 6){
	        Boolean result = false;
	        attempts = 0;
	        while (result == false){
	            System.out.println(arrayTreasure[choice-1].getQuestion());
	            answer = Integer.parseInt(scanner.nextLine());
	            result = arrayTreasure[choice-1].checkAnswer(answer);
	            attempts++;
	
	        }
	        System.out.println(arrayTreasure[choice-1].getPoints(attempts));
	
	    }
	}</pre>	

3(c)(v)	1 mark per screenshot
	• Screenshot:
	  outputting 2*2
	  entering 4
	  outputting 10
	
	• Screenshot:
	  outputting 3000+4000
	  entering an incorrect value
	  entering 7000
	  outputting 9
		2