

Cambridge International AS & A Level

COMPUTER SCIENCE**9618/42**

Paper 4 Practical

May/June 2026**MARK SCHEME**Maximum Mark: 75

Published

This mark scheme contains the instructions and marking guidance that our examiners used to mark candidate responses for this component.

Before they start marking, examiners complete a standardisation process. They review a range of candidate responses and discuss the mark scheme in detail to agree which answers can and cannot be accepted. Examiners award marks where it is clear that candidate responses have met the requirements of the mark scheme. These requirements may vary between different components or different versions of the same component, depending on the exact requirements of the question and the candidate responses seen during the standardisation process.

Sometimes, our mark schemes may allow marks to be awarded to responses which contain elements that are factually incorrect or for content not covered by the syllabus. We do this in response to the demands of a specific question to support candidates and make sure that our assessments are fair. However, these allowances should not be used as a guide for teaching and learning.

We cannot discuss the mark scheme with schools, and we recommend that schools read the mark scheme together with the assessment material and the Principal Examiner Report for Teachers.

This document consists of **41** printed pages.

General marking principles

All examiners must apply these marking principles when marking candidate responses.

1	Examiners must award marks for each response in line with: - the specific content defined in the mark scheme - the specific skills defined in the mark scheme or in the level descriptions - the standard of response required as exemplified by the standardisation scripts.
2	Examiners must award whole marks (not half marks, or other fractions).
3	Examiners must award marks positively . This means that: - marks are not deducted for errors or omissions - marks are awarded for correct/valid answers as defined in the mark scheme and also for other valid answers which go beyond the scope of the syllabus and mark scheme referring to your team leader as appropriate - the quality of spelling, punctuation and grammar are judged only when the mark scheme indicates that these features are specifically assessed in the question. However, the meaning of all responses must be unambiguous.
4	Examiners must consistently apply the requirements of the mark scheme and any rules for what to do when candidates have not followed instructions correctly.
5	Examiners must use the full range of marks defined in the mark scheme, unless limited by the quality of the candidate responses seen.
6	Examiners must award marks according to the mark scheme and must not award marks with grade thresholds or grade descriptions in mind.

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
	Benefit of the doubt
	To indicate where a key word/phrase/code is missing
	Incorrect
	Follow through
	Indicate a point in an answer
Highlighted text	To draw attention to a particular aspect or to indicate where parts of an answer have been combined
	Ignore
	Not answered question
	No benefit of doubt given
	No examples or not enough

Annotation	Meaning
	Not relevant or used to separate parts of an answer
Off-page comment	Allows comments to be entered at the bottom of the RM marking window and then displayed when the associated question item is navigated to.
	Repetition
	Indicates that work or a page has been seen including blank answer spaces and blank pages.
	Correct
	Too vague

PUBLISHED

- **Bold** in mark scheme means that idea is required.
- / in mark scheme means alternative.
- // in mark scheme means alternative solution that gains the same mark point.
- ... at the end of one mark point without a ... at the start of the next just means the sentence follows on. There is no dependency.
- ... at the end of one mark point and ... at the start of the next, this means the second cannot be awarded without the first.
- () means what is in the brackets is not required, or it is not required in some languages but may be required in others.

Question	Answer	Marks	Guidance
1(a)	1 mark each - Declaration of array with space for 9 strings (through declaration/initialisation) - Storing the correct data in the order given	2	
Example program code Python <pre>TheData = ["laptop", "tablet", "printer", "computer", "processor", "monitor", "memory", "cable", "touchscreen"]</pre> VB.NET <pre>Dim TheData(8) As String TheData(0) = "laptop" TheData(1) = "tablet" TheData(2) = "printer" TheData(3) = "computer" TheData(4) = "processor" TheData(5) = "monitor" TheData(6) = "memory" TheData(7) = "cable" TheData(8) = "touchscreen"</pre> Java <pre>String[] TheData = new String[9]; TheData[0] = "laptop"; TheData[1] = "tablet"; TheData[2] = "printer"; TheData[3] = "computer"; TheData[4] = "processor"; TheData[5] = "monitor"; TheData[6] = "memory"; TheData[7] = "cable"; TheData[8] = "touchscreen";</pre>			

Question	Answer	Marks	Guidance
1(b)(i)	1 mark each • Procedure header (and end where appropriate) taking one (string array) parameter • Looping through each string in the array • Finding the shortest string • Finding the longest string • Output the longest and shortest string in appropriate messages	5	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def FindLongestShortest(TheData): LongestLength = -1 LongestWord = "" ShortestLength = 999999999 ShortestWord = "" for x in range(len(TheData)): if len(TheData[x]) > LongestLength: LongestLength = len(TheData[x]) LongestWord = TheData[x] if len(TheData[x]) < ShortestLength: ShortestLength = len(TheData[x]) ShortestWord = TheData[x] print("The longest word is", LongestWord) print("The shortest word is", ShortestWord)</pre> VB.NET <pre>Sub FindLongestShortest(TheData()) Dim LongestLength As Integer = -1 Dim LongestWord As String = "" Dim ShortestLength As Integer = 999 Dim ShortestWord As String = "" For x = 0 To TheData.Length() - 1 If Len(TheData(x)) > LongestLength Then LongestLength = Len(TheData(x)) LongestWord = TheData(x) End If If Len(TheData(x)) < ShortestLength Then ShortestLength = Len(TheData(x)) ShortestWord = TheData(x) End If Next</pre>		

Question	Answer	Marks	Guidance
	<pre> Console.WriteLine("The longest word is " & LongestWord) Console.WriteLine("The shortest word is " & ShortestWord) End Sub Java public static void FindLongestShortest(String[] TheData){ Integer LongestLength = -1; String LongestWord = ""; Integer ShortestLength = 9999999; String ShortestWord = ""; for(Integer x = 0; x < TheData.length; x++){ if(TheData[x].length() > LongestLength){ LongestLength = TheData[x].length(); LongestWord = TheData[x]; } if((TheData[x]).length() < ShortestLength){ ShortestLength = TheData[x].length(); ShortestWord = TheData[x]; } } System.out.println("The longest word is " + LongestWord); System.out.println("The shortest word is " + ShortestWord); } </pre>		

Question	Answer	Marks	Guidance
1(b)(ii)	1 mark each calling FindLongestShortest with array as parameter	1	
Example program code Python <pre>FindLongestShortest(TheData)</pre> VB.NET <pre>FindLongestShortest(TheData)</pre> Java <pre>FindLongestShortest(TheData);</pre>			
1(b)(iii)	1 mark for screenshot of output	1	
Example output <pre>The longest word is touchscreen The shortest word is cable</pre>			
1(c)(i)	1 mark each - Function header (and end where appropriate) taking (string array) parameter - Looping through each element in the array... ...using insertion sort method e.g. taking element out and moving down - Using appropriate string comparison function/methods - Returning a correctly sorted array	5	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def InsertionSort(TheData): ArrayLength = len(TheData) if ArrayLength <= 1: return TheData for x in range(1, ArrayLength): Temp = TheData[x] y = x-1 while y >= 0 and Temp < TheData[y]: TheData[y+1] = TheData[y] y -= 1 TheData[y+1] = Temp return TheData</pre> VB.NET <pre>Function InsertionSort(TheData()) Dim ArrayLength As Integer = TheData.Length() Dim Temp As String Dim y As Integer If ArrayLength <= 1 Then Return TheData End If For x = 1 To ArrayLength - 1 Temp = TheData(x) y = x - 1 While y >= 0 If Temp < TheData(y) Then TheData(y + 1) = TheData(y) y = y - 1 Else Exit While End If End While TheData(y + 1) = Temp End For End Function</pre>		

Question	Answer	Marks	Guidance
	<pre> Next Return TheData End Function Java public static String[] InsertionSort(String[] TheData){ Integer ArrayLength = TheData.length; String Temp = ""; Integer y; if(ArrayLength <=1){ return TheData; } for(Integer x = 1; x < ArrayLength; x++){ Temp = TheData[x]; y = x - 1; while(y >= 0 && Temp.compareTo(TheData[y]) < 0){ TheData[y+1] = TheData[y]; y--; } TheData[y+1] = Temp; } return TheData; } </pre>		

Question	Answer	Marks	Guidance
1(c)(ii)	1 mark each • Outputting "Before sorting" and the array contents before sorting • Calling InsertionSort() with array and storing/using returned array • Outputting "After sorting" and the array contents after sorting	3	
Example program code Python <pre>print("Before Sorting") print(TheData) print("After sorting") TheData = InsertionSort(TheData) print(TheData)</pre> VB.NET <pre>Console.WriteLine("Before sorting") For x = 0 To 8 Console.WriteLine(TheData(x)) Next Console.WriteLine("After sorting") TheData = InsertionSort(TheData) For x = 0 To 8 Console.WriteLine(TheData(x)) Next</pre> Java <pre>System.out.println("Before sorting"); for(Integer x = 0; x < 9; x++){ System.out.println(TheData[x]); } System.out.println("After sorting"); TheData = InsertionSort(TheData); for(Integer x = 0; x < 9; x++){ System.out.println(TheData[x]); }</pre>			

PUBLISHED

Question	Answer	Marks	Guidance
1(c)(iii)	1 mark for output showing unsorted and then sorted array	1	FT data in array and output messages
<pre> Before Sorting ['laptop', 'tablet', 'printer', 'computer', 'processor', 'monitor', 'memory', 'cable', 'touchscreen'] After sorting ['cable', 'computer', 'laptop', 'memory', 'monitor', 'printer', 'processor', 'tablet', 'touchscreen'] </pre>			

PUBLISHED

Question	Answer	Marks	Guidance
2(a)(i)	1 mark each • Four private attributes with correct data types • Class <code>Avatar</code> header (and end where appropriate) • Constructor header (and end where appropriate) within class taking 4 parameters (Python 5) • Assigning parameters to attributes within constructor	4	

Question	Answer	Marks	Guidance
	Example program code Python <pre>class Avatar(): def __init__(self, pName, pGlasses, pHair, pAge): self.__TheName = pName #name as string self.__Glasses = pGlasses #glasses as boolean self.__Hair = pHair #hairColour as string self.__Age = pAge #age as integer</pre> VB.NET <pre>Class Avatar Private TheName As String Private Glasses As Boolean Private Hair As String Private Age As Integer Public Sub New(pName, pGlasses, pHair, pAge) TheName = pName Glasses = pGlasses Hair = pHair Age = pAge End Sub End Class</pre> Java <pre>class Avatar{ private String TheName; private boolean Glasses; private String Hair; private Integer Age; public Avatar(String pName, boolean pGlasses, String pHair, Integer pAge){ TheName = pName; Glasses = pGlasses; Hair = pHair; Age = pAge; } }</pre>		

Question	Answer	Marks	Guidance
2(a)(ii)	1 mark each • 1 get method header (and end where appropriate) with no parameter... • ... returning correct attribute • 2nd get method	3	
Example program code Python <pre>def GetName(self): return self.__TheName def GetAge(self): return self.__Age</pre> VB.NET <pre>Public Function GetName() Return TheName End Function Public Function GetAge() Return Age End Function</pre> Java <pre>public String GetName(){ return TheName; } public Integer GetAge(){ return Age; }</pre>			

Question	Answer	Marks	Guidance
2(a)(iii)	1 mark each - Method header (and end where appropriate) taking two parameters (Boolean and string) and returning a Boolean in all cases - Comparing parameters to correct attributes returning TRUE and FALSE correctly	2	Accept Meetcriteria

Example program code**Python**

```
def MeetCriteria(self, Glasses, Hair):
    if self.__Glasses == Glasses and Hair == self.__Hair:
        return True
    else:
        return False
```

VB.NET

```
Public Function MeetCriteria(pGlasses, pHair)
    If Glasses = pGlasses And Hair = pHair Then
        Return True
    Else
        Return False
    End If
End Function
```

Java

```
public boolean MeetCriteria(boolean pGlasses, String pHair){
    if(Glasses == pGlasses && Hair.equals(pHair)){
        return true;
    }else{
        return false;
    }
}
```

Question	Answer	Marks	Guidance
2(b)(i)	1 mark each • Procedure header (and end where appropriate) and appropriate exception handling with output • Opening text file (to read) and closing file • Looping until EOF/30 times... • ... reading in each line • Splitting each line by space • Instantiating object with correct values • Storing each object in appropriate global array	7	

Question	Answer	Marks	Guidance
Example program code Python	<pre>def LoadAvatars(): global AvailableAvatars try: TheFile = open("AvatarFile.txt") for Line in TheFile: DataSplit = (Line.strip()).split(" ") if DataSplit[1] == "True": DataSplit[1] = True else: DataSplit[1] = False AvailableAvatars.append(Avatar(DataSplit[0], bool(DataSplit[1]), DataSplit[2], int(DataSplit[3]))) TheFile.close() except: print("File not found") return AvailableAvatars</pre> VB.NET <pre>Public Sub LoadAvatars() Dim TempData() As String NumberAvatars = 0 Try Dim FileReader As New System.IO.StreamReader("AvatarFile.txt") While Not FileReader.EndOfStream TempData = (FileReader.ReadLine()).Split(" ") AvailableAvatars(NumberAvatars) = New Avatar(TempData(0), Convert.ToBoolean(TempData(1)), TempData(2), Convert.ToInt32(TempData(3))) NumberAvatars = NumberAvatars + 1 End While FileReader.Close() Catch ex As Exception Console.WriteLine("Cannot open or read from file") End Try End Sub</pre>		

Question	Answer	Marks	Guidance
	Java <pre> public static void LoadAvatars(){ String[] TempData = new String[4]; Integer NumberAvatars = 0; String FileName = "AvatarFile.txt"; try{ Scanner Scanner1 = new Scanner(new File(FileName)); while(Scanner1.hasNextLine()){ TempData = Scanner1.nextLine().split(" "); AvailableAvatars[NumberAvatars] = new Avatar(TempData[0], Boolean.parseBoolean(TempData[1]), TempData[2], Integer.parseInt(TempData[3])); NumberAvatars++; } Scanner1.close(); } catch (FileNotFoundException ex){ System.out.println("No file found"); } } </pre>		
2(b)(ii)	1 mark for calling LoadAvatars()	1	
	Example program code Python <pre>LoadAvatars()</pre> VB.NET <pre>LoadAvatars()</pre> Java <pre>LoadAvatars();</pre>		

Question	Answer	Marks	Guidance
2(c)(i)	1 mark each • Procedure header (and end where appropriate) and prompting for 2 inputs and reading 2 inputs ... • ... and validating appropriately • Looping through each avatar in the array ... • ... calling <code>MeetCriteria()</code> for each avatar and storing/using the return value • ... with correct values for glasses and hair • ... if return value is true, avatar is stored in a global array // reverse and delete	6	Accept <code>Meetcriteria</code>

Question	Answer	Marks	Guidance
Example program code Python	<pre>def EnterChoices(): global AvailableAvatars global OptionAvatars global NumberAvatars global OptionNumberAvatars Chosen = False Valid = False while Valid == False: GlassesChoice = input("Do you want them to wear glasses? ") if GlassesChoice.lower() == "yes": Valid = True TheGlasses = True elif GlassesChoice.lower() == "no": Valid = True TheGlasses = False else: print("Invalid, enter yes or no") HairChoice = -1 while HairChoice < 1 or HairChoice > 6: HairChoice = int(input("What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green ")) if HairChoice == 1: TheHair = "black" elif HairChoice == 2: TheHair = "blonde" elif HairChoice == 3: TheHair = "brown" elif HairChoice == 4: TheHair = "pink" elif HairChoice == 5: TheHair = "red" elif HairChoice == 6: TheHair = "green" for x in range(len(AvailableAvatars)): if AvailableAvatars[x].MeetCriteria(TheGlasses, TheHair): OptionAvatars.append(AvailableAvatars[x]) OptionNumberAvatars = OptionNumberAvatars + 1</pre>		

Question	Answer	Marks	Guidance
VB.NET	<pre> Public Sub EnterChoices() Dim Chosen As Boolean = False Dim Valid As Boolean = False Dim GlassesChoice As String = "" Dim TheGlasses As String = "" Dim HairChoice As Integer Dim TheHair As String = "" While Valid = False Console.WriteLine("Do you want them to wear glasses") GlassesChoice = Console.ReadLine() If GlassesChoice = "yes" Then Valid = True TheGlasses = True ElseIf GlassesChoice = "no" Then Valid = True TheGlasses = False Else Console.WriteLine("Invalid, enter yes or no") End If End While HairChoice = -1 While HairChoice < 1 Or HairChoice > 6 Console.WriteLine("What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green") HairChoice = Console.ReadLine() If HairChoice = 1 Then TheHair = "black" ElseIf HairChoice = 2 Then TheHair = "blonde" ElseIf HairChoice = 3 Then TheHair = "brown" ElseIf HairChoice = 4 Then TheHair = "pink" </pre>		

Question	Answer	Marks	Guidance
	<pre> ElseIf HairChoice = 5 Then TheHair = "red" ElseIf HairChoice = 6 Then TheHair = "green" End If End While OptionNumberAvatars = 0 For x = 0 To 29 If AvailableAvatars(x).MeetCriteria(TheGlasses, TheHair) Then OptionAvatars(OptionNumberAvatars) = AvailableAvatars(x) OptionNumberAvatars = OptionNumberAvatars + 1 End If Next End Sub Java public static void EnterChoices(){ Scanner scanner = new Scanner(System.in); System.out.println("Do you want them to wear glasses?"); String GlassesChoice = scanner.nextLine(); boolean Choice = false; boolean Valid = false; boolean TheGlasses = false; Integer HairChoice = -1; String TheHair = ""; while(Valid == false){ if(GlassesChoice.equals("yes")){ Valid = true; TheGlasses = true; }else if(GlassesChoice.equals("no")){ Valid = true; TheGlasses = false; </pre>		

Question	Answer	Marks	Guidance
	<pre> }else{ System.out.println("Invalid, enter yes or no"); } } HairChoice = -1; while(HairChoice < 1 HairChoice > 6){ System.out.println("What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green"); HairChoice = Integer.parseInt(scanner.nextLine()); if(HairChoice == 1){ TheHair = "black"; }else if(HairChoice == 2){ TheHair = "blonde"; }else if(HairChoice == 3){ TheHair = "brown"; }else if(HairChoice == 4){ TheHair = "pink"; }else if(HairChoice == 5){ TheHair = "red"; }else if(HairChoice == 6){ TheHair = "green"; } } OptionNumberAvatars = 0; for(Integer x = 0; x < 30; x++){ if(AvailableAvatars[x].MeetCriteria(TheGlasses, TheHair)){ OptionAvatars[OptionNumberAvatars] = AvailableAvatars[x]; OptionNumberAvatars++; } } } } } </pre>		

Question	Answer	Marks	Guidance
2(c)(ii)	1 mark each • Calling <code>EnterChoices()</code> • Counting/calculating the number of avatars and outputting • Looping through each avatar that meets the criteria ... • ...outputting the name and age in an appropriate message ... • ...using <code>get</code> methods.	5	
Example program code Python <pre> EnterChoices() print("There are", len(OptionAvatars), "avatars that meet your criteria") if len(OptionAvatars) > 0: for x in range(len(OptionAvatars)): print(OptionAvatars[x].GetName(), "who is", OptionAvatars[x].GetAge(), "years old") </pre> VB.NET <pre> EnterChoices() Console.WriteLine("There are" & OptionNumberAvatars & "avatars that meet your criteria") If OptionNumberAvatars > 0 Then For x = 0 To OptionNumberAvatars - 1 Console.WriteLine(OptionAvatars(x).GetName() & " who is " & OptionAvatars(x).GetAge() & " years old") Next End If </pre> Java <pre> EnterChoices(); System.out.println("There are" + OptionNumberAvatars + " avatars that meet your criteria"); if(OptionNumberAvatars > 0){ for(Integer x = 0; x < OptionNumberAvatars; x++){ System.out.println(OptionAvatars[x].GetName() + " who is " + OptionAvatars[x].GetAge() + " years old"); } } </pre>			

Question	Answer	Marks	Guidance
2(d)(i)	1 mark for inputs and output For blue accept any invalid data	1	FT appropriate message
Do you want them to wear glasses? yes What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green 7 What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green 1 There are 4 avatars that meet your criteria Dax who is 22 years old Pia who is 22 years old Rei who is 18 years old Ren who is 19 years old			
2(d)(ii)	1 mark for inputs and output of 0 avatars	1	FT appropriate message
Do you want them to wear glasses? no What colour hair? 1 for black, 2 for blonde, 3 for brown, 4 for pink, 5 for red, 6 for green 6 There are 0 avatars that meet your criteria			

Question	Answer	Marks	Guidance
3(a)	1 mark each • <code>LinkedList</code> 2D array with 8×2 elements (integer global) • <code>StartLinkedList</code> (int) initialised with 2 • <code>StartEmptyList</code> (int) initialised with 7 • <code>LinkedList</code> initialised with correct data ... • ... in correct indices	5	

Question	Answer	Marks	Guidance
	Example program code Python <pre> LinkedList = [] LinkedList.append([9512,5]) LinkedList.append([5541,3]) LinkedList.append([1999,0]) LinkedList.append([6651,-1]) LinkedList.append([-1,6]) LinkedList.append([5612,1]) LinkedList.append([-1,-1]) LinkedList.append([-1,4]) StartLinkedList = 2 StartEmptyList = 7 </pre> VB.NET <pre> Dim LinkedList(7, 1) As Integer Dim StartLinkedList As Integer Dim StartEmptyList As Integer LinkedList(0, 0) = 9512 LinkedList(0, 1) = 5 LinkedList(1, 0) = 5541 LinkedList(1, 1) = 3 LinkedList(2, 0) = 1999 LinkedList(2, 1) = 0 LinkedList(3, 0) = 6651 LinkedList(3, 1) = -1 LinkedList(4, 0) = -1 LinkedList(4, 1) = 6 LinkedList(5, 0) = 5612 LinkedList(5, 1) = 1 LinkedList(6, 0) = -1 LinkedList(6, 1) = -1 LinkedList(7, 0) = -1 LinkedList(7, 1) = 4 StartLinkedList = 2 StartEmptyList = 7 </pre>		

Question	Answer	Marks	Guidance
	Java <pre> public static Integer[][] LinkedList = new Integer[8][2]; public static Integer StartLinkedList; public static Integer StartEmptyList; LinkedList[0][0] = 9512; LinkedList[0][1] = 5; LinkedList[1][0] = 5541; LinkedList[1][1] = 3; LinkedList[2][0] = 1999; LinkedList[2][1] = 0; LinkedList[3][0] = 6651; LinkedList[3][1] = -1; LinkedList[4][0] = -1; LinkedList[4][1] = 6; LinkedList[5][0] = 5612; LinkedList[5][1] = 1; LinkedList[6][0] = -1; LinkedList[6][1] = -1; LinkedList[7][0] = -1; LinkedList[7][1] = 4; StartLinkedList = 2; StartEmptyList = 7; </pre>		

Question	Answer	Marks	Guidance
3(b)(i)	1 mark each • Procedure header (and end where appropriate) and starting at <code>StartLinkedList</code> • Following pointers • ... outputting data of each node • Stopping when empty node found // stopping when null pointer	4	
Example program code Python <pre>def PrintListItems(): global LinkedList global StartEmptyList global StartLinkedList TempPosition = StartLinkedList while LinkedList[TempPosition][1] != -1: print(LinkedList[TempPosition][0]) TempPosition = LinkedList[TempPosition][1] print(LinkedList[TempPosition][0])</pre> Alternative Python: <pre>def PrintListItems(): global StartLinkedList, LinkedList node = StartLinkedList while node != -1: print(LinkedList[node][0]) node = LinkedList[node][1]</pre> VB.NET <pre>Sub PrintListItems() Dim TempPosition As Integer = StartLinkedList While (LinkedList(TempPosition, 1) <> -1) Console.WriteLine(LinkedList(TempPosition, 0)) TempPosition = LinkedList(TempPosition, 1) End While Console.WriteLine(LinkedList(TempPosition, 0)) End Sub</pre>			

Question	Answer	Marks	Guidance
Java	<pre>public static void PrintListItems(){ Integer TempPosition = StartLinkedList; while(LinkedList[TempPosition][1] != -1){ System.out.println(LinkedList[TempPosition][0]); TempPosition = LinkedList[TempPosition][1]; } System.out.println(LinkedList[TempPosition][0]); }</pre>		

Question	Answer	Marks	Guidance
3(b)(ii)	1 mark each • Procedure header and starting at <code>StartEmptyList</code> • Following pointers and outputting data • Stopping correctly	3	
Example program code Python <pre>def PrintEmptyList(): global LinkedList global StartEmptyList TempPosition = StartEmptyList Flag = True while Flag == True: if LinkedList[TempPosition][1] == -1: Flag = False print(TempPosition) TempPosition = LinkedList[TempPosition][1]</pre> VB.NET <pre>Sub PrintEmptyList() Dim TempPosition = StartEmptyList Dim Flag As Boolean = True While Flag = True If LinkedList(TempPosition, 1) = -1 Then Flag = False End If Console.WriteLine(TempPosition) TempPosition = LinkedList(TempPosition, 1) End While End Sub</pre>			

Question	Answer	Marks	Guidance
	Java <pre> public static void PrintEmptyList(){ Integer TempPosition = StartEmptyList; boolean Flag = true; while(Flag == true){ if(LinkedList[TempPosition][1] == -1){ Flag = false; } System.out.println(TempPosition); TempPosition = LinkedList[TempPosition][1]; } } </pre>		
3(b)(iii)	1 mark each • Outputting Linked list then calling <code>PrintListItems()</code> • Outputting Empty list then calling <code>PrintEmptyList()</code>	2	
	Example program code Python <pre> print("Linked list") PrintListItems() print("Empty list") PrintEmptyList() </pre> VB.NET <pre> Console.WriteLine("Linked list") PrintListItems() Console.WriteLine("Empty list") PrintEmptyList() </pre> Java <pre> System.out.println("Linked list"); PrintListItems(); System.out.println("Empty list"); PrintEmptyList(); </pre>		

Question	Answer	Marks	Guidance
3(b)(iv)	1 mark for output of linked list data 1 mark for output of empty list data	2	
<pre> Linked list 1999 9512 5612 5541 6651 Empty list 7 4 6 </pre>			

Question	Answer	Marks	Guidance
3(c)(i)	1 mark each to max 8 • Procedure header taking one (integer) parameter • Checking if list is full (e.g. <code>if StartEmptyList != -1</code>) ... • ...and outputting "List is full" • Storing node in <code>StartEmptyList</code> index ... • ...with pointer <code>-1</code> (may be done at separate position in algorithm) • Starting at first node (<code>StartLinkedList</code>) ... • ...following pointers until empty node • Changing last pointer to <code>StartEmptyList</code> (or temp) • Updating <code>StartEmptyList</code> to previous (could be temp) node's pointer	8	

Question	Answer	Marks	Guidance
	Example program code Python <pre>def AddItem(Item): global LinkedList global StartEmptyList global StartLinkedList TempPosition = 0 if StartEmptyList != -1: LinkedList[StartEmptyList][0] = Item NewEmptyList = LinkedList[StartEmptyList][1] LinkedList[StartEmptyList][1] = -1 TempPosition = StartLinkedList while LinkedList[TempPosition][1] != -1: TempPosition = LinkedList[TempPosition][1] LinkedList[TempPosition][1] = StartEmptyList StartEmptyList = NewEmptyList else: print("List is full")</pre> VB.NET <pre>Sub AddItem(ByVal Item) Dim TempPosition As Integer = 0 Dim NewEmptylist As Integer = 0 If StartEmptyList <> -1 Then LinkedList(StartEmptyList, 0) = Item NewEmptylist = LinkedList(StartEmptyList, 1) LinkedList(StartEmptyList, 1) = -1 TempPosition = StartLinkedList While (LinkedList(TempPosition, 1) <> -1) TempPosition = LinkedList(TempPosition, 1) End While LinkedList(TempPosition, 1) = StartEmptyList StartEmptyList = NewEmptylist Else Console.WriteLine("List is full") End If End Sub</pre>		

Question	Answer	Marks	Guidance
	<pre> Java public static void AddItem(Integer Item){ Integer TempPosition = 0; Integer NewEmptyList = 0; if(StartEmptyList != -1){ LinkedList[StartEmptyList][0] = Item; NewEmptyList = LinkedList[StartEmptyList][1]; LinkedList[StartEmptyList][1] = -1; TempPosition = StartLinkedList; while(LinkedList[TempPosition][1] != -1){ TempPosition = LinkedList[TempPosition][1]; } LinkedList[TempPosition][1] = StartEmptyList; StartEmptyList = NewEmptyList; }else{ System.out.println("List is full"); } } </pre>		

Question	Answer	Marks	Guidance
3(c)(ii)	1 mark each • Calling <code>AddItem()</code> with parameter 8451 • Calling both output procedures with suitable headings	2	
Example program code Python <pre>AddItem(8451) print("Linked list") PrintListItems() print("Empty list") PrintEmptyList()</pre> VB.NET <pre>AddItem(8451) Console.WriteLine("Linked list") PrintListItems() Console.WriteLine("Empty list") PrintEmptyList()</pre> Java <pre>AddItem(8451); System.out.println("Linked list"); PrintListItems(); System.out.println("Empty list"); PrintEmptyList();</pre>			

Question	Answer	Marks	Guidance
3(c)(iii)	1 mark for output showing updated linked list contents and empty list	1	
<pre> add item Linked list 1999 9512 5612 5541 6651 8451 Empty list 4 6 </pre>			