CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/42**

Paper 4 Practical **May/June 2024**

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 May/June 2024 series for most Cambridge IGCSE, Cambridge International A and AS Level and Cambridge Pre-U components, and some Cambridge O Level components.

This document consists of **50** printed pages.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions 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.

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(a)</td><td>

1 mark each to max 6

- Procedure declaration (and end where appropriate) taking (string) parameter
- Declaration of array to store the data read (type string, suitable number of elements e.g. 150)
- Opening file to read…
- ... using exception handling with try and catch and output
- Reading in the data for each line in that file and storing in array…
- … removing carriage return (Java, Python)
- Counting the number of words
- Closing the file (might be within the Python opening file statement)

e.g.

Java

```
public static void ReadWords(String FileName){


   try{
      FileReader f = new FileReader(FileName);

   try{
      BufferedReader Reader = new BufferedReader(f);
      String Line= Reader.readLine();

      while (Line != null){

         WordArray[NumberWords] = Line.replace("\n","");
         NumberWords++;
         Line = Reader.readLine();

      }

      Reader.close();

   }catch(IOException ex){}
```

</td><td><b>6</b></td></tr>
</table>

Question	Answer	Marks
1(a)		

```
}catch(FileNotFoundException e){
   System.out.println("File not found");
}

}
```

VB.NET

```
Sub ReadWords(FileName As String)

   Try
      Dim DataReader As StreamReader = New StreamReader(FileName)
      NumberWords = 0
      Do Until DataReader.EndOfStream
         WordArray(NumberWords) = DataReader.ReadLine()
         NumberWords = NumberWords + 1
      Loop

      DataReader.Close()
   Catch ex As Exception
      Console.WriteLine("Invalid file")
   End Try

End Sub
```

Python

```
def ReadWords(FileName):
   global WordArray
   global NumberWords
   File = open(FileName, 'r')
   DataRead = File.read().strip()
   File.close()
   WordArray = DataRead.split()
   NumberWords = len(WordArray)
```

Question	Answer	Marks
1(b)	1 mark each • Outputting message to ask user to enter easy, medium, hard • Taking input from user • Conversion of input to filename… • … calling `ReadWords()` with correct filename in each case e.g.	**4**

Java

```
public static void main(String args[]){
   NumberWords = 0;
   Scanner scanner = new Scanner(System.in);
   System.out.println("Easy, medium or hard?");
   String Choice = scanner.nextLine();
   if(Choice.equals("Easy")){
      ReadWords("Easy.txt");
   }else if(Choice.equals("medium")){
      ReadWords("Medium.txt");
   }else{
      ReadWords("Hard.txt");
   }
}
```

VB.NET

```
Sub Main(args As String())
   Console.WriteLine("Easy, medium or hard?")
   Dim FileName As String
   Dim Choice As String = Console.ReadLine().ToLower()
   If Choice = "easy" Then
      FileName = "Easy.txt"
   ElseIf Choice = "medium" Then
      FileName = "Medium.txt"
   Else
```

Question	Answer	Marks

1(b)	`      FileName = "Hard.txt"`
	`   End If`
	
	`   ReadWords(FileName)`
	`End Sub`
	
	Python
	
	`WordArray = []`
	`NumberWords = 0`
	`Choice = input("Easy, medium or hard? ").lower()`
	`if Choice == "easy":`
	`   File = "Easy.txt"`
	`elif Choice == "medium":`
	`   File = "Medium.txt"`
	`else:`
	`   File = "Hard.txt"`
	`ReadWords(File)`	

Question	Answer	Marks
1(c)(i)	1 mark each • Procedure (and end) taking array and number of answers as parameters **and** outputting the main word and the number of answers • Loops until user requests to stop (enters "no") …. • … takes word as input and compares input to each answer in array but **not** the main word • … method of recording answers found e.g. replaces with "" (or appropriate null) • … outputs if found and not found • Counts the number of answers found (in loop, second array, any method) e.g. Java	**6**

```
public static void Play(){
   System.out.println(NumberWords);
   Scanner scanner = new Scanner(System.in);
   String WordChosen = WordArray[0];
   System.out.println("The word is " + WordChosen);
   System.out.println("There are " + NumberWords + " words that can be made with 3 or more
letters");
   WordArray[0] = "";
   Boolean Contin = true;
   Integer QuantityFound = 0;
   String WordInput;
   Boolean Found = false; String Answer = "yes";

   while(!(Answer.equals("no"))){

   System.out.println("Enter your word or no to stop");
   Answer = scanner.nextLine();
   Found = false;
   if(!(Answer.equals("no"))){
   for(Integer x = 0; x <= NumberWords; x++){
      if(Answer.equals(WordArray[x])){
         WordArray[x] = "";
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(c)(i)</td><td><pre>
         QuantityFound++;
         System.out.println("Correct, you have found " + QuantityFound + " words");
         Found = true;
      }
   }
   if(Found == false){
      System.out.println("Sorry that was incorrect");
   }
   }

   }

}
</pre>
VB.NET
<pre>
Sub Play()
   Dim Word As String = WordArray(0)
   Console.WriteLine("The word is: " & Word)
   Console.WriteLine("There are " & NumberWords & " words that can be made with 3 or more
letters")
   WordArray(0) = ""
   Dim Contin As Boolean = True
   Dim QuantityFound As Integer = 0
   Dim Found As Boolean
   Dim Answer As String = "yes"
   While Answer <> "no"
      Console.WriteLine("Enter your word or no to stop")
      Answer = Console.ReadLine().ToLower()

   Found = False
      If Answer <> "Not" Then
         For x = 0 To NumberWords
            If Answer = WordArray(x) Then
</pre></td><td></td></tr>
</table>

Question	Answer	Marks
1(c)(i)		

```
               WordArray(x) = ""
               QuantityFound = QuantityFound + 1
               Console.WriteLine("Correct, you have found " & QuantityFound & " words")
               Found = True
               x = NumberWords + 1
            End If

         Next x
         If Found = False Then
            Console.WriteLine("Sorry that was incorrect")
         End If

      End If

   End While

End Sub
```

Python

```
def Play():
   global WordArray
   global NumberWords
   Word = WordArray[0]
   print("The word is: ", Word)
   print("There are", NumberWords-1,"words that can be made with 3 or more letters")
   WordArray[0] = ""

   Answer = "yes"
   QuantityFound = 0
   while Answer != "no":
      Answer = input("Enter your word or no to stop ").lower()
      Found = False
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>1(c)(i)</td><td><pre>
if Answer != "no":
   for x in range(0, NumberWords):
      if Answer == WordArray[x]:

         WordArray[x] = ""
         QuantityFound = QuantityFound + 1
         print("Correct, you have found", QuantityFound, "words")
         Found = True
      if Found == False:
         print("Sorry that was incorrect")
</pre></td><td></td></tr>
</table>

Question	Answer	Marks
1(c)(ii)	1 mark each • Calculates and outputs percentage of answers found (when ‘no’ is entered) • Method of identifying answers not found (e.g. looping array and skipping null values)… • … and outputting those answers	**3**

e.g.

Java

```
   public static void Play(){
      System.out.println(NumberWords);
      Scanner scanner = new Scanner(System.in);
      String WordChosen = WordArray[0];
      System.out.println("The word is " + WordChosen);

      System.out.println("There are " + NumberWords + " words that can be made with 3 or
more letters");
      WordArray[0] = "";
      Boolean Contin = true;
      Integer QuantityFound = 0;
      String WordInput;
      Boolean Found = false;
      String Answer = "yes";

      while(!(Answer.equals("no"))){
         System.out.println("Enter your word or no to stop");
         Answer = scanner.nextLine();
         Found = false;
         if(!(Answer.equals("no"))){
            for(Integer x = 0; x <= NumberWords; x++){
               if(Answer.equals(WordArray[x])){
                  WordArray[x] = "";
                  QuantityFound++;
                  System.out.println("Correct, you have found " + QuantityFound + "
words");
```

Question	Answer	Marks
1(c)(ii)		

```
                  Found = true;
               }
            }
            if(Found == false){
               System.out.println("Sorry that was incorrect");
            }
            }

      }
      double Correct = ((Double.valueOf(QuantityFound) / Double.valueOf(NumberWords)) *
100.0);
      System.out.println("You found " + Correct + "%");

      if(Correct < 100){

         System.out.println("The words you missed are");
         for(Integer x = 0; x <= NumberWords; x++){

            if(WordArray[x] != ""){
               System.out.println(WordArray[x]);
            }

         }

      }

}
```

VB.NET

```
Sub Play()
   Dim Word As String = WordArray(0)
   Console.WriteLine("The word is: " & Word)
   Console.WriteLine("There are " & NumberWords & " words that can be made with 3 or more
letters")
```

Question	Answer	Marks
1(c)(ii)		

```
WordArray(0) = ""
Dim Contin As Boolean = True
Dim QuantityFound As Integer = 0
Dim Found As Boolean
Dim Answer As String = "yes"
While Answer <> "no"
   Console.WriteLine("Enter your word or no to stop")
   Answer = Console.ReadLine().ToLower()

   Found = False
   If Answer <> "Not" Then
      For x = 0 To NumberWords
         If Answer = WordArray(x) Then

            WordArray(x) = ""
            QuantityFound = QuantityFound + 1
            Console.WriteLine("Correct, you have found " & QuantityFound & " words")
            Found = True
            x = NumberWords + 1
         End If

      Next x
         If Found = False Then
            Console.WriteLine("Sorry that was incorrect") End If
         End If

End While

Dim Correct As Double
Correct = (QuantityFound / NumberWords) * 100
Console.WriteLine("You found " & Correct & "%")
If Correct < 100 Then
   Console.WriteLine("The words you missed are ")
   For x = 0 To NumberWords
      If WordArray(x) <> "" Then
```

Question	Answer	Marks
1(c)(ii)		

```
            Console.WriteLine(WordArray(x))
         End If
      Next x
   End If
End Sub
```

Python

```
def Play():
   global WordArray
   global NumberWords
   Word = WordArray[0]
   print("The word is: ", Word)
   print("There are", NumberWords-1,"words that can be made with 3 or more letters")
   WordArray[0] = ""
   Answer = "yes"
   QuantityFound = 0

   while Answer != "no":
      Answer = input("Enter your word or no to stop ").lower()
      Found = False
      if Answer != "no":
         for x in range(0, NumberWords):
            if Answer == WordArray[x]:

               WordArray[x] = ""
               QuantityFound = QuantityFound + 1
               print("Correct, you have found", QuantityFound, "words")
               Found = True
            if Found == False:
               print("Sorry that was incorrect")
   Correct = (QuantityFound / (NumberWords-1)) * 100
   print("You found", Correct,"%")
```

Question	Answer	Marks

1(c)(ii)	`if Correct < 100:`
	`   print("The words you missed are")`
	`   for x in range(0, NumberWords-1):`
	`      if WordArray[x] != "":`
	`         print(WordArray[x])`	

Question	Answer	Marks
1(d)(i)	1 mark for: • Calling `Play()` with array and number of answers after all read in from file	**1**

e.g.

Java

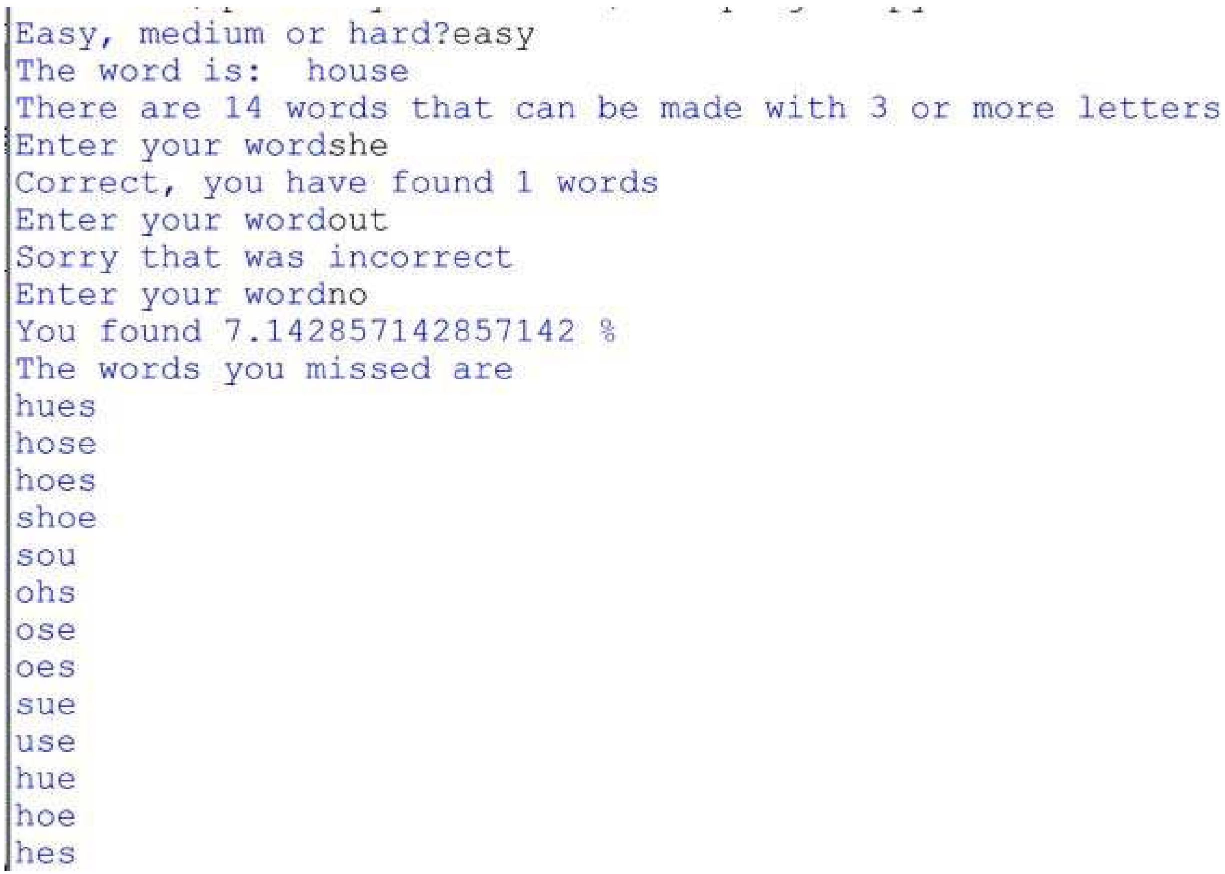
```
public static void ReadWords(String FileName){

    try{
        FileReader f = new FileReader(FileName);

        try{

            BufferedReader Reader = new BufferedReader(f);
            String Line= Reader.readLine();

            while (Line != null){

                WordArray[NumberWords] = Line.replace("\n","");
                NumberWords++;
                Line = Reader.readLine();

            }

            Reader.close();

            Play();

        }catch(IOException ex){}

            }catch(FileNotFoundException e){
                System.out.println("File not found");
            }
        }
```

Question	Answer	Marks
1(d)(i)	VB.NET	

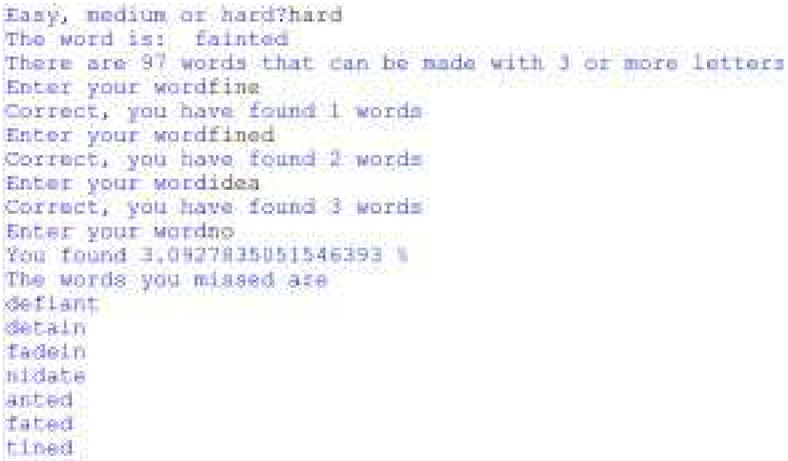
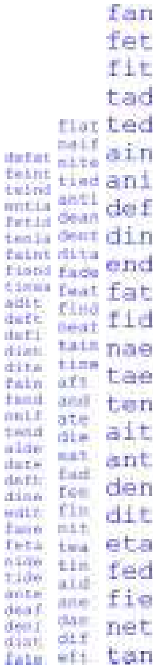
```
Sub ReadWords(FileName As String)

   Try
      Dim DataReader As StreamReader = New StreamReader(FileName)
      NumberWords = 0
      Do Until DataReader.EndOfStream
         WordArray(NumberWords) = DataReader.ReadLine()
         NumberWords = NumberWords + 1
      Loop

      DataReader.Close()
      Play()
   Catch ex As Exception
      Console.WriteLine("Invalid file")
   End Try
End Sub
```

Python

```
def ReadWords(FileName):
   global WordArray
   global NumberWords
   File = open(FileName, 'r')
   DataRead = File.read().strip()
   File.close()
   WordArray = DataRead.split()
   NumberWords = len(WordArray)
   Play()
```

Question	Answer	Marks
1(d)(ii)	1 mark for screenshot showing the inputs "easy", "she", "out", "no" e.g. 	1

```
Easy, medium or hard?easy
The word is:  house
There are 14 words that can be made with 3 or more letters
Enter your wordshe
Correct, you have found 1 words
Enter your wordout
Sorry that was incorrect
Enter your wordno
You found 7.142857142857142 %
The words you missed are
hues
hose
hoes
shoe
sou
ohs
ose
oes
sue
use
hue
hoe
hes
```

Question	Answer	Marks
1(d)(iii)	1 mark for screenshot showing the inputs ‘hard’, ‘fine’, ‘fined’, ‘idea’, ‘no’ e.g.  	**1**

```
Easy, medium or hard?hard
The word is:  fainted
There are 97 words that can be made with 3 or more letters
Enter your wordfine
Correct, you have found 1 words
Enter your wordfined
Correct, you have found 2 words
Enter your wordidea
Correct, you have found 3 words
Enter your wordno
You found 3.0927835051546393 %
The words you missed are
defiant
detain
fadein
nidate
anted
fated
tined
```

```
defat
feint
teind
entia
fetid
tenia
faint
fiend
tinea
adit
daft
defi
diet
dita
fain
fand
naif
tend
aide
date
deft
dine
edit
fane
feta
nide
tide
ante
deaf
deni
dint
fate
```

```
fiat
naif
nite
tied
anti
dean
dent
dita
fade
feat
find
neat
tain
tine
aft
and
ate
die
eat
fad
fen
fin
nit
tea
tin
aid
ane
dae
dif
eft
```

```
fan
fet
fit
tad
ted
ain
ani
def
din
end
fat
fid
nae
tae
ten
ait
ant
den
dit
eta
fed
fie
net
tan
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(a)(i)</td><td>

1 mark each to max 4

- Class declaration (and end where appropriate) with identifier `Node`
- `LeftPointer`, `Data` and `RightPointer`, integer
- Constructor taking 1 parameter (within class) …
- … assigning parameter to `Data` initialising `LeftPointer` and `RightPointer` to –1

e.g.

Java

```
public class Node{
   private Integer LeftPointer;
   private Integer Data;
   private Integer RightPointer;

   public Node(Integer PData){
      LeftPointer = -1;
      Data = PData;
      RightPointer = -1;
   }
}
```

VB.NET

```
Class Node
   Private LeftPointer As Integer
   Private Data As Integer
   Private RightPointer As Integer
   Sub New(PData)
      LeftPointer = -1
      Data = PData
      RightPointer = -1
   End Sub
End Class
```

</td><td>4</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(a)(i)</td><td>Python<pre>
class Node():
   def __init__(self, PData):
      self._LeftPointer = -1 #int
      self._Data = PData #int
      self.__RightPointer = -1 #int
</pre></td><td></td></tr>
</table>

Question	Answer	Marks
2(a)(ii)	1 mark each • 1 get method with no parameter… • …returning correct attribute • Remaining 2 correct (FT minor errors)	3

e.g.

Java

```
public Integer GetLeft(){
   return LeftPointer;
   }
   public Integer GetRight(){
      return RightPointer;
   }
   public Integer GetData(){
      return Data;
   }
```

VB.NET

```
Function GetLeft()
   Return LeftPointer
End Function
Function GetRight()
   Return RightPointer
End Function
Function GetData()
   Return Data
End Function
```

Python

```
def GetLeft(self):
   return self._LeftPointer
```

Question	Answer	Marks

2(a)(ii)	`def GetRight(self):`
	`    return self._RightPointer`
	`def GetData(self):`
	`    return self.__Data`	

Question	Answer	Marks
2(a)(iii)	1 mark each • 1 set method with parameter … • … assigning to attribute • Remaining 2 correct (FT minor errors) e.g.	**3**

Java

```
public void SetLeft(Integer NewLeft){
   LeftPointer = NewLeft;
}
public void SetRight(Integer NewRight){
   RightPointer = NewRight;
}
public void SetData(Integer NewData){
   Data = NewData;
}
```

VB.NET

```
Sub SetLeft(NewLeft)
   LeftPointer = NewLeft
End Sub
Sub SetRight(NewRight)
   RightPointer = NewRight
End Sub
Sub SetData(NewData)
   Data = NewData
End Sub
```

Python

```
def SetLeft(self, NewLeft):
   self._LeftPointer = NewLeft
def SetRight(self, NewRight):
```

Question	Answer	Marks

2(a)(iii)	`    self._RightPointer = NewRight`
	`def SetData(self, NewData):`
	`    self.__Data = NewData`	

Question	Answer	Marks
2(b)(i)	1 mark each • Class header (and end) • Private array `Tree` of type `Node` with 20 elements, private `FirstNode` and private `NumberNodes` • Constructor assigns –1 to `FirstNode` and 0 to `NumberNodes` • … initialises all `Tree` (20) elements to `Node` object with data value –1	**4**

e.g.

Java

```
class TreeClass{

   private static Node[] Tree = new Node[20];
   private static Integer FirstNode;
   private static Integer NumberNodes;

   public TreeClass(){
      FirstNode = -1;
      NumberNodes = 0;
      Integer MinusOne = -1;
      for(Integer x = 0; x < 20; x++){
         Tree[x] = new Node(MinusOne);
      }
   }
}
```

VB.NET

```
Class TreeClass
   Private Tree(20) As Node
   Private FirstNode As Integer
   Private NumberNodes As Integer

   Sub New()
      FirstNode = -1
      NumberNodes = 0
```

Question	Answer	Marks

2(b)(i)	`      For x = 0 To 19`
	`         Tree(x) = New Node(-1)`
	`      Next`
	`   End Sub`
	`End Class`
	
	Python
	
	`class TreeClass():`
	
	`   def __init__(self):`
	`      self.__Tree = [] #type node 20 spaces`
	`      self._FirstNode = -1 #int`
	`      self._NumberNodes = 0 #int`
	`      for x in range(20):`
	`         self.__Tree.append(Node(-1))`	

Question	Answer	Marks
2(b)(ii)	1 mark each: • Method header and end, taking node as parameter **and** checking if empty and inserting in first position, updating `FirstNode` • … otherwise inserting node in tree • Accessing first node and comparing data … • … checking whether to go left or right … • … repeatedly until data found • Updating left and right pointer for parent node	**6**

e.g.

Java

```
public void InsertNode(Node NewNode){
   Integer NodeAccess;
   Integer Previous = -1;
   String Direction;

   if(NumberNodes == 0){

      Tree[0] = NewNode;
      FirstNode = 0;
      NumberNodes++;
   }else{

      Tree[NumberNodes] = NewNode;
      NodeAccess = FirstNode;
      Direction = "";
      System.out.println(Tree[0].GetData());
      while(NodeAccess != -1){
         Previous = NodeAccess;

         if(NewNode.GetData() < Tree[NodeAccess].GetData()){
            NodeAccess = Tree[NodeAccess].GetLeft();
            Direction = "left";
```

Question | Answer | Marks

2(b)(ii)

```
         }else if(NewNode.GetData() > Tree[NodeAccess].GetData()){
            NodeAccess = Tree[NodeAccess].GetRight();
            Direction = "right";
         }

      }
      if(Direction.equals("left")){
         Tree[Previous].SetLeft(NumberNodes);
      }else{
         Tree[Previous].SetRight(NumberNodes);
      }
      NumberNodes++;
   }
}
```

VB.NET

```
Sub InsertNode(NewNode)
   Dim NodeAccess As Integer
   Dim Direction As String
   Dim Previous As Integer

   If NumberNodes = 0 Then
      Tree(0) = NewNode
      FirstNode = 0
      NumberNodes += 1
   Else
      Tree(NumberNodes) = NewNode
      NodeAccess = FirstNode
      Direction = ""

      While NodeAccess <> -1
         Previous = NodeAccess
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(b)(ii)</td><td>

```
         If NewNode.GetData() < Tree(NodeAccess).GetData() Then
            NodeAccess = Tree(NodeAccess).GetLeft()
            Direction = "left"
         ElseIf NewNode.GetData() > Tree(NodeAccess).GetData() Then
            NodeAccess = Tree(NodeAccess).GetRight()
            Direction = "right"
         End If
      End While

      If Direction = "left" Then
         Tree(Previous).SetLeft(NumberNodes)
      Else
         Tree(Previous).SetRight(NumberNodes)
      End If
      NumberNodes += 1
   End If
End Sub
```

Python

```
def InsertNode(self, NewNode):

   if(self.__NumberNodes == 0):

      self._Tree[0] = NewNode self._FirstNode = 0
      self._NumberNodes = self._NumberNodes + 1
   else:
      self.__Tree[self.__NumberNodes] = NewNode

      NodeAccess = self._FirstNode
      Direction = ""

      while(NodeAccess != -1):
         Previous = NodeAccess
         if NewNode.GetData() < self.__Tree[NodeAccess].GetData():
```

</td><td></td></tr>
</table>

Question	Answer	Marks
2(b)(ii)		

```
         NodeAccess = self._Tree[NodeAccess].GetLeft()
         Direction = "left"
      elif NewNode.GetData() > self.__Tree[NodeAccess].GetData():

         NodeAccess = self._Tree[NodeAccess].GetRight()
         Direction = "right"

   if(Direction == "left"):

      self._Tree[Previous].SetLeft(self._NumberNodes)
   else:
      self._Tree[Previous].SetRight(self._NumberNodes)
      self._NumberNodes = self._NumberNodes + 1
```

Question	Answer	Marks
2(b)(iii)	1 mark each • Procedure header (and end) with no parameter and if no nodes output ‘No nodes’ • (otherwise) Loop from index 0 to NumberNodes (or equivalent) … • … Outputting LeftPointer, Data then RightPointer • … using get methods	4

e.g.

Java

```
public void OutputTree(){
   if(NumberNodes == 0){
      System.out.println("No nodes");
   }else{
      for(Integer x = 0; x < NumberNodes; x++){
         System.out.println(Tree[x].GetLeft() + " " + Tree[x].GetData() + " " +
Tree[x].GetRight());

      }
   }
}
```

VB.NET

```
Sub OutputTree()
   If NumberNodes = 0 Then
      Console.WriteLine("No nodes")
   Else
      For x = 0 To NumberNodes - 1
         Console.WriteLine(Tree(x).GetLeft() & " " & Tree(x).GetData() & " " &
Tree(x).GetRight())
      Next
   End If
End Sub
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(b)(iii)</td><td>

Python

```
def OutputTree(self):
   if self._NumberNodes == 0:
      print("No nodes")
   else:
      for x in range(0, self.__NumberNodes):
         print(self.__Tree[x].GetLeft(), " ", self.__Tree[x].GetData(), " ",self.
Tree[x].GetRight())
```

</td><td></td></tr>
<tr><td>2(c)(i)</td><td>

1 mark for

- Instance of `TreeClass` created with identifier `TheTree`

e.g.

Java

```
public static void main(String args[]){
   TreeClass TheTree = new TreeClass();
}
```

VB.NET

```
Sub Main(args As String())
   Dim TheTree As TreeClass = New TreeClass()
End Sub
```

Python

```
TheTree = TreeClass()
```

</td><td>1</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>2(c)(ii)</td><td>1 mark each
• Creating one node with one correct value (e.g. 10)
• Calling <code>InsertNode</code> for <code>TheTree</code> with each new <code>Node</code>
• All nodes correctly assigned in order
• Calling <code>OutputTree</code>

e.g.

Java
<pre>
public static void main(String args[]){

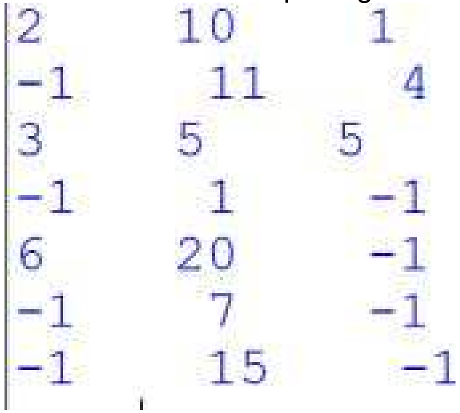
   TreeClass TheTree = new TreeClass();

   TheTree.InsertNode(new Node(10));
   TheTree.InsertNode(new Node(11));
   TheTree.InsertNode(new Node(5));
   TheTree.InsertNode(new Node(1));
   TheTree.InsertNode(new Node(20));
   TheTree.InsertNode(new Node(7));
   TheTree.InsertNode(new Node(15));

   TheTree.OutputTree();
}
</pre>VB.NET
<pre>
Sub Main(args As String())
   Dim TheTree As TreeClass = New TreeClass()
   TheTree.InsertNode(New Node(10))
   TheTree.InsertNode(New Node(11))
   TheTree.InsertNode(New Node(5))
   TheTree.InsertNode(New Node(1))
   TheTree.InsertNode(New Node(20))
   TheTree.InsertNode(New Node(7))
   TheTree.InsertNode(New Node(15))
   TheTree.OutputTree()
End Sub
</pre></td><td>4</td></tr>
</table>

Question	Answer	Marks

2(c)(ii)	Python
	
	`TheTree = TreeClass()`
	`TheTree.InsertNode(Node(10))`
	`TheTree.InsertNode(Node(11))`
	`TheTree.InsertNode(Node(5))`
	`TheTree.InsertNode(Node(1))`
	`TheTree.InsertNode(Node(20))`
	`TheTree.InsertNode(Node(7))`
	`TheTree.InsertNode(Node(15))`
	`TheTree.OutputTree()`	

2(c)(iii)	1 mark for correct output e.g.
	
	`2    10    1`
	`-1    11    4`
	`3    5    5`
	`-1    1    -1`
	`6    20    -1`
	`-1    7    -1`
	`-1    15    -1`
		1

Question	Answer	Marks
3(a)	1 mark each • `NumberArray` declared (in main) with the 7 correct values in order (integer) 100 85 644 22 15 8 1	**1**

e.g.

Java

```
public static void main(String args[]){
   Integer[] NumberArray = new Integer[7];
   NumberArray[0] = 100;
   NumberArray[1] = 85;
   NumberArray[2] = 644;
   NumberArray[3] = 22;
   NumberArray[4] = 15;
   NumberArray[5] = 8;
   NumberArray[6] = 1;
}
```

VB.NET

```
Sub Main(args As String())
   Dim NumberArray(7) As Integer
   NumberArray(0) = 100
   NumberArray(1) = 85
   NumberArray(2) = 644
   NumberArray(3) = 22
   NumberArray(4) = 15
   NumberArray(5) = 8
   NumberArray(6) = 1
EndSub
```

Python

```
NumberArray = [100, 85, 644, 22, 15, 8, 1]
```

Question	Answer	Marks
3(b)(i)	1 mark each • Recursive function written with recursive call • Correct base case and return • Correct while loop control and internal • All correct and structure followed e.g. Java	**4**

```
public static Integer[] RecursiveInsertion(Integer[] IntegerArray, Integer NumberElements){
   Integer LastItem;
   Integer CheckItem;

   if(NumberElements <= 1){
      return IntegerArray;
   }else{
      RecursiveInsertion(IntegerArray, NumberElements - 1);
      LastItem = IntegerArray[NumberElements - 1];
      CheckItem = NumberElements - 2;
   }
   Boolean LoopAgain = true;
   if(CheckItem < 0){
      LoopAgain = false;
   }else if(IntegerArray[CheckItem] < LastItem){
      LoopAgain = false;
   }

   while(LoopAgain){
      IntegerArray[CheckItem + 1] = IntegerArray[CheckItem];
      CheckItem = CheckItem - 1;
      if(CheckItem < 0){
         LoopAgain = false;
```

Question	Answer	Marks
3(b)(i)		

```
      }else if(IntegerArray[CheckItem] <= LastItem){
         LoopAgain = false;
      }
   }
   IntegerArray[CheckItem + 1] = LastItem;
   return IntegerArray;
}
```

VB.NET

```
Function RecursiveInsertion(IntegerArray, NumberElements)
   Dim LastItem, CheckItem As Integer
   If NumberElements <= 1 Then
      Return IntegerArray
   Else
      RecursiveInsertion(IntegerArray, NumberElements - 1)
      LastItem = IntegerArray(NumberElements - 1)
      CheckItem = NumberElements - 2
   End If

   Dim LoopAgain As Boolean = True
   If CheckItem < 0 Then
      LoopAgain = False
   ElseIf IntegerArray(CheckItem) <= LastItem Then
      LoopAgain = False
   End If

   While LoopAgain

      IntegerArray(CheckItem + 1) = IntegerArray(CheckItem)
      CheckItem = CheckItem - 1

      If CheckItem < 0 Then
         LoopAgain = False
      ElseIf IntegerArray(CheckItem) <= LastItem Then
```

Question	Answer	Marks
3(b)(i)		

```
            LoopAgain = False
        End If
    End While

    IntegerArray(CheckItem + 1) = LastItem
    Return IntegerArray

End Function
```

Python

```
def RecursiveInsertion(IntegerArray, NumberElements):

    if NumberElements <= 1:
        return IntegerArray

    RecursiveInsertion(IntegerArray,NumberElements - 1)
    LastItem = IntegerArray[NumberElements - 1]
    CheckItem = NumberElements - 2

    LoopAgain = True
    if CheckItem < 0:
        LoopAgain = False
    elif IntegerArray[CheckItem] <= LastItem:
        LoopAgain = False

    while (LoopAgain):
        IntegerArray[CheckItem + 1] = IntegerArray[CheckItem]
            CheckItem = CheckItem - 1

        if CheckItem < 0:
            LoopAgain = False
        elif IntegerArray[CheckItem] <= LastItem:
            LoopAgain = False

    IntegerArray[CheckItem + 1] = LastItem return IntegerArray
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(b)(ii)</td><td>

1 mark each

- Calling `RecursiveInsertion()` with array and number of elements (7 or length)
- Outputting ‘recursive’ and then each element in **returned** array

e.g.

Java

```
Integer[] SortedArray = new Integer[7];
SortedArray = RecursiveInsertion(NumberArray, 7);
System.out.println("Recursive");
for(Integer x = 0; x < 7; x++){
   System.out.println(SortedArray[x]);
}
```

VB.NET

```
SortedArray = RecursiveInsertion(NumberArray, 7)
Console.WriteLine("Recursive")
For x = 0 To 6
   Console.WriteLine(SortedArray(x))
Next x
```

Python

```
SortedArray = RecursiveInsertion(NumberArray, len(NumberArray))
print("Recursive", SortedArray)
```

</td><td>2</td></tr>
</table>

Question	Answer	Marks
3(b)(iii)	1 mark for screenshot with: Recursive 1 8 15 22 85 100 644	**1**

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(c)(i)</td><td>
1 mark each
• Insertion algorithm written with correct identifier – no recursion
• External loop while there are still elements left (e.g. NumberElements > 0)
• Internal loop and selection accurate
• Nothing additional added / logic changed

e.g.

Java
<pre>
public static Integer[] IterativeInsertion(Integer[] IntegerArray, Integer NumberElements){
   Integer LastItem;
   Integer CheckItem;
   while(NumberElements > 0){
      LastItem = IntegerArray[NumberElements - 1];
      CheckItem = NumberElements - 2;

      Boolean LoopAgain = true;
      if(CheckItem < 0){
         LoopAgain = false;
      }else if(IntegerArray[CheckItem] < LastItem){
         LoopAgain = false;
      }

      while(LoopAgain){
         IntegerArray[CheckItem + 1] = IntegerArray[CheckItem];
         CheckItem = CheckItem - 1;
         if(CheckItem < 0){
            LoopAgain = false;
         }else if(IntegerArray[CheckItem] <= LastItem){
            LoopAgain = false;
         }
      }
</pre>
</td><td>4</td></tr>
</table>

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(c)(i)</td><td>
<pre>
        IntegerArray[CheckItem + 1] = LastItem;
        NumberElements = NumberElements - 1;
    }
    return IntegerArray;
}
</pre>
VB.NET
<pre>
Function IterativeInsertion(IntegerArray, NumberElements)
    Dim LastItem, CheckItem As Integer
    While NumberElements > 0
        LastItem = IntegerArray(NumberElements - 1)
        CheckItem = NumberElements - 2
        Dim LoopAgain As Boolean = True
        If CheckItem < 0 Then
            LoopAgain = False
        ElseIf IntegerArray(CheckItem) <= LastItem Then
            LoopAgain = False
        End If
        While LoopAgain
            IntegerArray(CheckItem + 1) = IntegerArray(CheckItem)
            CheckItem = CheckItem - 1
            If CheckItem < 0 Then
                LoopAgain = False
            ElseIf IntegerArray(CheckItem) <= LastItem Then
                LoopAgain = False
            End If
        End While
        IntegerArray(CheckItem + 1) = LastItem
        NumberElements = NumberElements - 1

    End While
    Return IntegerArray
End Function
</pre>
</td><td></td></tr>
</table>

Question	Answer	Marks
3(c)(i)	Python	

```
def IterativeInsertion(IntegerArray, NumberElements):
   while NumberElements > 0:
      LastItem = IntegerArray[NumberElements - 1]
      CheckItem = NumberElements - 2
      LoopAgain = True
      if CheckItem < 0:
         LoopAgain = False
      elif IntegerArray[CheckItem] <= LastItem:
         LoopAgain = False
      while(LoopAgain):
         IntegerArray[CheckItem + 1] = IntegerArray[CheckItem]
         CheckItem = CheckItem - 1
         if CheckItem < 0:
            LoopAgain = False
         elif IntegerArray[CheckItem] <= LastItem:
            LoopAgain = False

         IntegerArray[CheckItem + 1] = LastItem
         NumberElements = NumberElements - 1
      return IntegerArray
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(c)(ii)</td><td>

1 mark each

- Calling `IterativeInsertion()` with original unsorted array and outputting 'iterative' and the content of the **returned** array

e.g.

Java

```
Integer[] Sorted2Array = new Integer[7];
Sorted2Array = IterativeInsertion(NumberArray, 7);
System.out.println("iterative");
for(Integer x = 0; x < 7; x++){
   System.out.println(Sorted2Array[x]);
}
```

VB.NET

```
Sorted2Array = IterativeInsertion(NumberArray, 7)
Console.WriteLine("iterative")
For x = 0 To 6
   Console.WriteLine(Sorted2Array(x))
Next x
```

Python

```
Sorted2Array = IterativeInsertion(NumberArray, len(NumberArray))
print("iterative", Sorted2Array)
```

</td><td><b>1</b></td></tr>
<tr><td>3(c)(iii)</td><td>

1 mark for Recursive
1 8 15 22 85 100 644

Iterative
1 8 15 22 85 100 644

</td><td><b>1</b></td></tr>
</table>

Question	Answer	Marks
3(d)(i)	1 mark each to max 6	**6**

- Recursive function `BinarySearch` taking the 4 indicated parameters
- Suitable base case (e.g. First > Last) …
- … returning –1
- Calculating middle element
- Comparing `ToFind` with middle and returning Middle if found
- If `ToFind` less than middle, recursive call with Last as Middle – 1
- If `ToFind` more than middle, recursive call with First as Middle + 1

e.g.

Java

```
public static Integer BinarySearch(Integer[] IntegerArray, Integer First, Integer Last,
Integer ToFind){
   Integer Middle;
   if(First > Last){;
      return -1;
   }else{
      Middle = (Last + First) / 2;

      if(IntegerArray[Middle].equals(ToFind)){
         return Middle;
      }else if(IntegerArray[Middle] > ToFind){
         return BinarySearch(IntegerArray, First, Middle - 1, ToFind);
      }else{
         return BinarySearch(IntegerArray, Middle + 1, Last, ToFind);
      }
   }
}
```

VB.NET

```
Function BinarySearch(IntegerArray, First, Last, ToFind)
   Dim Middle As Integer
   If First > Last Then
```

Question	Answer	Marks
3(d)(i)		

```
      Return -1
   Else
      Middle = (Last + First) \ 2
      If IntegerArray(Middle) = ToFind Then

         Return Middle
      ElseIf IntegerArray(Middle) > ToFind Then
         Return BinarySearch(IntegerArray, First, Middle - 1, ToFind)
      Else
         Return BinarySearch(IntegerArray, Middle + 1, Last, ToFind)
      End If
   End If
End Function
```

Python

```
def BinarySearch(IntegerArray, First, Last, ToFind):

   if First > Last:
      return -1
   else:
      Middle = int((Last + First) / 2)

      if IntegerArray[Middle] == ToFind:
         return Middle

      elif IntegerArray[Middle] > ToFind:
         return BinarySearch(IntegerArray, First, Middle - 1, ToFind)

      else:
         return BinarySearch(IntegerArray, Middle + 1, Last, ToFind)
```

<table>
<tr><th>Question</th><th>Answer</th><th>Marks</th></tr>
<tr><td>3(d)(ii)</td><td>

1 mark each

- Calling `BinarySearch` function with sorted array, 0, 6/len(array)–1, 644 as parameters
- Checking return value and outputting ‘Not found’ if –1 and index otherwise

e.g.

Java

```
Position = BinarySearch(Sorted2Array, 0, 6, 644);
if(Position == -1){
   System.out.println("Not found");
}else{
   System.out.println(Position);
}
```

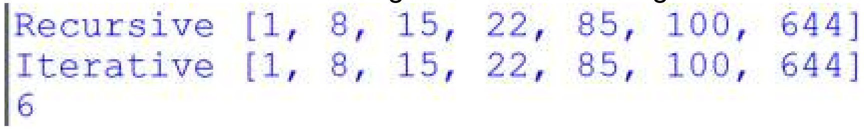
VB.NET

```
Position = BinarySearch(Sorted2Array, 0, 6, 644)
If Position = -1 Then
   Console.WriteLine("Not found")
Else
   Console.WriteLine(Position)
End If
```

Python

```
Position = BinarySearch(Sorted2Array, 0, len(NumberArray)-1, 644)
if Position == -1:
   print("Not found")
else:
   print(Position)
```

</td><td>2</td></tr>
</table>

Question	Answer	Marks
3(d)(iii)	1 mark for screenshot showing found in index 6 e.g.  Recursive [1, 8, 15, 22, 85, 100, 644] Iterative [1, 8, 15, 22, 85, 100, 644] 6	**1**