

|      |                                                                                                                                                                                                                                                                                                                                              |   |                                                                                       |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|---------------------------------------------------------------------------------------|
| 3(a) | Mark as follows:                                                                                                                                                                                                                                                                                                                             |   |                                                                                       |
|      | MP1                                                                                                                                                                                                                                                                                                                                          | 1 | open the file <b>OldFile.txt</b> in <b>read (mode)</b>                                |
|      | MP2                                                                                                                                                                                                                                                                                                                                          | 2 | open the file <b>NewFile.txt</b> in <b>write (mode)</b>                               |
|      |                                                                                                                                                                                                                                                                                                                                              | 3 | read a line from <b>OldFile.txt</b> and store in <b>LineX</b>                         |
|      | MP3                                                                                                                                                                                                                                                                                                                                          | 4 | read a line from <b>OldFile.txt</b> and store in <b>LineY</b>                         |
|      |                                                                                                                                                                                                                                                                                                                                              | 5 | read a line from <b>OldFile.txt</b> and store in <b>LineZ</b>                         |
|      | MP4                                                                                                                                                                                                                                                                                                                                          | 6 | set <b>NewString</b> to<br><b>LineX &amp; '\ ' &amp; LineY &amp; '\ ' &amp; LineZ</b> |
|      | MP5                                                                                                                                                                                                                                                                                                                                          | 7 | write <b>NewString</b> to <b>NewFile.txt</b>                                          |
|      | MP6                                                                                                                                                                                                                                                                                                                                          | 8 | repeat from step 3 until <b>end of OldFile.txt</b>                                    |
|      |                                                                                                                                                                                                                                                                                                                                              | 9 | close both files                                                                      |
| 3(b) | It does not appear in (any of) the data items<br>// it does not appear in the ID <b>and</b> Name                                                                                                                                                                                                                                             |   |                                                                                       |
| 3(c) | <b>MP1</b> Add a ( <b>single</b> ) new separator <b>at the end of the third (original) item</b><br><b>MP2</b> Calculate the <b>length</b> of <b>both new data items</b><br><b>MP3</b> Store / append the <b>length</b> (of the new each items)<br><b>MP4</b> as a fixed <b>length</b> / separated digit string // by example<br><b>Max 3</b> |   |                                                                                       |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8(a) | <p>Example solution:</p> <pre> PROCEDURE CountLoans(ThisStudentID : STRING)   DECLARE Index, LCount, RCount : INTEGER    LCount ← 0   RCount ← 0    FOR Index ← 1 TO 7000     IF Loan[Index].StudentID = ThisStudentID THEN       IF Loan[Index].OnLoan = FALSE THEN         RCount ← RCount + 1       ELSE         LCount ← LCount + 1       ENDIF     ENDIF   NEXT Index    OUTPUT "Student has ", LCount, " books on loan"   OUTPUT "and has returned ", RCount, " books"  ENDPROCEDURE </pre> <p>Mark as follows:</p> <p><b>MP1</b> Declaration of Index <b>and</b> two count variables</p> <p><b>MP2</b> Loop for 7000 iterations</p> <p><b>MP3</b> Index references an <b>indexed data</b> item with correct <b>.dot notation</b></p> <p><b>MP4</b> Attempt to compare the <b>StudentID</b> field = <b>parameter in a loop</b></p> <p><b>MP5</b> Test of the <b>OnLoan</b> field <b>and</b> correct logic for both outcomes</p> <p><b>MP6</b> Both count variables initialised <b>and</b> increment appropriate count <b>in a loop</b></p> <p><b>MP7</b> Output includes both counts with a <u>suitable</u> message</p> |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8(b) | <p><b>Example solution:</b></p> <pre> FUNCTION NewLoan(ThisStudentID, ThisBookID : STRING) RETURNS     BOOLEAN     DECLARE Index : INTEGER     DECLARE IsStored : BOOLEAN     CONSTANT Blank = ""      Index ← 1     IsStored ← FALSE      WHILE Index &lt;= 7000 AND NOT IsStored         IF Loan[Index].StudentID = Blank THEN             Loan[Index].StudentID ← ThisStudentID             Loan[Index].BookID ← ThisBookID             Loan[Index].OnLoan ← TRUE             IsStored ← TRUE         ENDIF         Index ← Index + 1     ENDWHILE      RETURN IsStored ENDFUNCTION </pre> <p>Mark as follows:</p> <p><b>MP1</b> Loop through 7000 elements in Loan array – correct loop structure<br/> <b>MP2</b> or until the first unused element is found // Correct RETURN inside a FOR loop<br/> <b>MP3</b> Correct use of the indexed dot notation<br/> <b>MP4</b> Test if record is unused (.StudentID = "") in a loop<br/> <b>MP5</b> Two item assignment statements correct<br/> <b>MP6</b> Loan[Index].OnLoan assigned TRUE<br/> <b>MP7</b> Correct logic to return BOOLEAN in both cases (position found / no position available)</p> <p><b>Alternative solution: loop only (no 'found' mechanism)</b></p> <pre> FOR Index ← 1 TO 7000     IF Loan[Index].StudentID = Blank THEN         Loan[Index].StudentID ← ThisStudentID         Loan[Index].BookID ← ThisBookID         Loan[Index].OnLoan ← TRUE         RETURN TRUE     ENDIF NEXT RETURN FALSE </pre> |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8(c) | <p><b>MP1</b> Save to/Open a (text) file<br/> <b>MP2</b> <b>Single line</b> – form a string with separator characters and write to the file // <b>Multiple lines</b> - write each data item to a separate line in the file</p>                                                                                                                                                                                                                                                                |
| 8(d) | <p><b>Additional date(s) solution</b></p> <p><b>MP1</b> Store the <b>return date</b> of each loan // the <b>start date</b> <u>and</u> the <b>loan period</b><br/> <b>MP2</b> Algorithm will do a <b>calculation</b> involving the MP1 data item(s)</p> <p>OR</p> <p><b>The student email address solution</b></p> <p><b>MP3</b> Store the student <u>email address</u><br/> <b>MP4</b> In order <b>to send</b> the email // the student is alerted to return the book</p> <p><b>Max 2</b></p> |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8(a) | <p><b>Example solution:</b></p> <pre> PROCEDURE LoanStatus(ThisStudentID, ThisBookID : STRING)     DECLARE Index : INTEGER     DECLARE Found : BOOLEAN      Index ← 1 / 0     Found ← FALSE      WHILE Index &lt;= 8000 / 7999 AND NOT Found         IF Loan[Index].StudentID = ThisStudentID AND             Loan[Index].BookID = ThisBookID THEN             IF Loan[Index].OnLoan = FALSE THEN                 OUTPUT "Loan has been returned."             ELSE                 OUTPUT "Loan has not been returned."             ENDIF             Found ← TRUE         ENDIF         Index ← Index + 1     ENDWHILE      IF NOT Found THEN         OUTPUT "Warning - Loan not found."     ENDIF ENDPROCEDURE </pre> <p>Mark as follows:</p> <ol style="list-style-type: none"> <li>1 Procedure heading, parameters and ending</li> <li>2 Loop through <b>all elements</b> in Loan array</li> <li>3 ... or until record loan is found</li> <li>4 Test for correct StudentID <b>in a loop</b></li> <li>5 ... and correct BookID</li> <li>6 ... check OnLoan data item</li> <li>7 ... and output <u>both</u> messages as appropriate regarding book loan status <b>once</b></li> <li>8 If no matching loan found, Output 'not found' <b>after the loop</b></li> </ol> |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8(b)      | <p>Example solution:</p> <pre> FUNCTION LoansPerTutor(TutorID : STRING) RETURNS INTEGER   DECLARE Index, Count : INTEGER    Count ← 0    FOR Index ← 1 / 0 TO 8000 / 7999     IF LEFT(Loan[Index].StudentID, 3) = TutorID AND       Loan[Index].OnLoan = TRUE THEN       Count ← Count + 1     ENDIF   NEXT Index    RETURN Count ENDFUNCTION </pre> <p>Mark as follows:</p> <ol style="list-style-type: none"> <li>1 Initialise local integer for Count</li> <li>2 Loop through <b>all elements</b> in the Loan array</li> <li>3 Attempt at referencing a data item <b>in a loop</b></li> <li>4 Extract TutorID from Loan[Index].StudentID <b>in a loop</b></li> <li>5 Test for Matching Tutor <b>AND</b> test if the book has been returned using correct dot notation <b>in a loop</b></li> <li>6 ... if true then increment Count <b>in a loop</b></li> <li>7 Return count</li> </ol> |
| 8(c)(i)   | <p>One mark per point:</p> <ol style="list-style-type: none"> <li>1 Include (a suffix string based on) the date of the archive / the next number in sequence</li> <li>2 Concatenate with a root filename such as 'Archive'</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 8(c)(ii)  | <p>Boolean data cannot be written to a text file // OnLoan is not a string // OnLoan will have to be converted to text</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 8(c)(iii) | <p>One mark per point:</p> <p>Benefit: Algorithm to <b>store / extract</b> a record is easier // unpacking of data is not required // No need for string concatenation</p> <p>Drawback: More file <b>accesses needed</b> (to read / write a complete record)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2(a) | <p>1 mark each</p> <ul style="list-style-type: none"> <li>• Queue declared as a (global) 1D array of 100 string elements all initialised to ""</li> <li>• QueueHead and QueueTail initialised with -1, NumberItems initialised to 0</li> </ul> <p>Example program code</p> <p>Java</p> <pre> public static String[] Queue = new String[100]; public static Integer QueueHead; public static Integer QueueTail; public static Integer NumberItems; for(int x = 0; x &lt; 100; x++){   Queue[x] = ""; } QueueHead = -1; QueueTail = -1; NumberItems = 0; </pre> <p>VB.NET</p> <pre> Dim Queue(99) As String Dim QueueHead As Integer Dim QueueTail As Integer Dim NumberItems As Integer For x = 0 To 99   Queue(x) = "" Next QueueHead = -1 QueueTail = -1 NumberItems = 0 </pre> |
| 2(a) | <p>Python</p> <pre> global Queue, QueueHead, QueueTail, NumberItems Queue = [] for x in range(100):   Queue.append("") QueueHead = -1 QueueTail = -1 NumberItems = 0 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

2(b)

1 mark each

- Function header (and end where appropriate) taking one (string) parameter **and** checking full **and** returning FALSE
- (otherwise) Storing parameter in QueueTail + 1
- Incrementing QueueTail and NumberItems
- (Dealing with first element) If QueueHead = -1 store 0 in QueueHead / increment QueueHead
- Return TRUE in all cases when inserted

Example program code.

Java

```
public static Boolean Enqueue(String TheData){
    if(QueueHead == -1){
        Queue[0] = TheData;
        QueueHead = 0;
        QueueTail = 0;
        NumberItems++;
        return true;
    }else if(QueueTail >= 99){
        return false;
    }else{
        Queue[QueueTail+1] = TheData;
        QueueTail++;
        NumberItems++;
        return true;
    }
}
```

2(b)

VB.NET

```
Function Enqueue(TheData)
    If QueueHead = -1 Then
        Queue(0) = TheData
        QueueHead = 0
        QueueTail = 0
        NumberItems += 1
        Return True
    ElseIf QueueTail >= 99 Then
        Return False
    Else
        Queue(QueueTail + 1) = TheData
        QueueTail += 1
        NumberItems += 1
        Return True
    End If
End Function
```

Python

```
def Enqueue(TheData):
    global Queue, QueueHead, QueueTail, NumberItems
    if(QueueHead == -1){
        Queue[0] = TheData
        QueueHead = 0
        QueueTail = 0
        NumberItems +=1
        return True
    elif QueueTail >= 99:
        return False
    else:
        Queue[QueueTail+1] = TheData
        QueueTail +=1
        NumberItems +=1
        return True
```

2(c)

1 mark each

- Function header (and end), checking if Queue is empty (NumberItems = 0 or QueueHead > QueueTail) **and** returning "False"
- (otherwise) returning Queue[QueueHead]
- Incrementing QueueHead, decrementing NumberItems

Example program code

Java

```
public static String Dequeue(){
    String ReturnValue;
    if(NumberItems == 0){
        return "False";
    }else{
        ReturnValue = Queue[QueueHead];
        QueueHead++;
        NumberItems--;
        return ReturnValue;
    }
}
```

VB.NET

```
Function Dequeue()

    If NumberItems = 0 Then
        Return "False"
    Else
        Dequeue = Queue(QueueHead)
        QueueHead += 1
        NumberItems -= 1
    End If
End Function
```

2(c)

Python

```
def Dequeue():
    global Queue, QueueHead, QueueTail, NumberItems
    if NumberItems == 0:
        return "False"
    else:
        ReturnData = Queue[QueueHead]
        QueueHead += 1
        NumberItems -=1
        return ReturnData
```

- 2(d) 1 mark each to max 5
- Procedure header (and end where appropriate)
  - Opening the file **and** closing the file (in appropriate place)
  - Looping until EOF // looping through each line ...
  - ... read in each value ...
  - ... calling Enqueue() with read in value and store/use return value
  - Exception handling with try except and appropriate output with all file access within try

## Example program code

## Java

```
public static void ReadData(){
    Boolean ReturnValue;
    Boolean FinishLoop = false;
    try{
        Scanner Scanner1 = new Scanner(new File("BinaryData.txt"));
        while(Scanner1.hasNextLine() && FinishLoop == false){
            ReturnValue = Enqueue(Scanner1.nextLine());
            if(ReturnValue.equals("False")){
                FinishLoop = true;
            }
        }
        Scanner1.close();
    } catch(FileNotFoundException ex){
        System.out.println("No file found");
    }
}
```

- 2(e) 1 mark each
- Procedure header (and end where appropriate), storing final string compressed data in global variable
  - Call Dequeue() **and** storing return value ...
  - ... repeatedly until the return value == "False"
  - Comparing new value with previous ...
  - ... maintaining counter of number of occurrences ...
  - ... appending digit and number of occurrences to a string without overriding

## Example program code

## Java

```
public static void Compress(){
    String First = Dequeue();
    String NewLine = "";
    Integer Count;
    String NextChar;
    while(NumberItems > 0 && First != "False"){
        Count = 1;
        NextChar = Dequeue();
        while(NextChar.equals(First)){
            Count++;
            First = NextChar;
            NextChar = Dequeue();
        }
        NewLine = First + Count;
        NewString = NewString + NewLine;
        First = NextChar;
    }
}
```

2(d) **VB.NET**

```
Sub ReadData()
    Dim ReturnValue As String
    Dim DataReader As New System.IO.StreamReader("BinaryData.txt")
    Dim FinishLoop As Boolean = True
    Do Until DataReader.EndOfStream Or FinishLoop = False
        ReturnValue = Enqueue(DataReader.ReadLine())
        If ReturnValue = False Then
            FinishLoop = False
        End If
    Loop
    DataReader.Close()
End Sub
```

## Python

```
def ReadData():
    TheFile = open("BinaryData.txt")
    for Line in TheFile:
        ReturnValue = Enqueue(Line.strip())
        if ReturnValue == False:
            break

    TheFile.close()
```

2(e) **VB.NET**

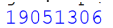
```
Sub Compress()
    Dim First As String = Dequeue()
    Dim NewLine As String = ""
    Dim Count As Integer
    Dim NextChar As String
    While NumberItems > 0 And First <> "False"
        Count = 1
        NextChar = Dequeue()

        While NextChar = First
            Count += 1
            First = NextChar
            NextChar = Dequeue()
        End While

        NewLine = First & Count
        NewString = NewString & NewLine
        First = NextChar
    End While
```

## Python

```
def Compress():
    global NewString
    First = Dequeue()
    NewString = ""
    while NumberItems > 0 and First != "False":
        Count = 1
        NextChar = Dequeue()
        while NextChar == First:
            Count += 1
            First = NextChar
            NextChar = Dequeue()
        NewLine = First + str(Count)
        NewString = NewString + NewLine
        First = NextChar
```

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2(f)(i)  | <p>1 mark each</p> <ul style="list-style-type: none"> <li>• Calling ReadData() then Compress()</li> <li>• Outputting compressed string</li> </ul> <p>Example program code</p> <p><b>Java</b></p> <pre>public static void main(String args[]){     NewString = "";     for(int x = 0; x &lt; 100; x++){         Queue[x] = "";     }     QueueHead = -1;     QueueTail = -1;     NumberItems = 0;      ReadData();     Compress();     System.out.println(NewString); }</pre> <p><b>VB.NET</b></p> <pre>Sub Main()     NewString = ""     For x = 0 To 99         Queue(x) = ""     Next     QueueHead = -1     QueueTail = -1     NumberItems = 0      ReadData()     Compress()     Console.WriteLine(NewString)     Console.ReadLine() End Sub</pre> |
| 2(f)(i)  | <p><b>Python</b></p> <pre>NewString = "" Queue = [] for x in range(100):     Queue.append("") QueueHead = -1 QueueTail = -1 NumberItems = 0 ReadData() Compress() print(NewString)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 2(f)(ii) | <p>1 mark for screenshot showing output</p> <p>Example</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(a) | <p>Example solution:</p> <pre>FUNCTION CustomerOrder(Number, Points : INTEGER) RETURNS                                                     INTEGER      DECLARE NewPoints, NumberFree: INTEGER      NewPoints ← Points + Number     NumberFree ← 0     WHILE NewPoints &gt;= 11 AND Number &gt; 0         NumberFree ← NumberFree + 1         NewPoints ← NewPoints -11         Number ← Number - 1     END WHILE      OUTPUT "Number of free coffees is: ", NumberFree     RETURN NewPoints ENDFUNCTION</pre> <p><b>MP1</b> Function header <b>and</b> ending <b>and</b> parameters <b>and</b> return type</p> <p><b>MP2</b> Number of coffees ordered added to points</p> <p><b>MP3</b> Correct calculation of one free coffee</p> <p><b>MP4</b> Attempted calculation of multiple free coffees <b>and</b> reduced NewPoints</p> <p><b>MP5</b> Output number of free drinks with suitable message</p> <p><b>MP6</b> Return of NewPoints</p> |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(b)(i) | <p><b>Example solution:</b></p> <pre> PROCEDURE AddNewCustomers (NumToAdd : INTEGER)      DECLARE CustomerID : INTEGER     DECLARE Count : INTEGER     DECLARE Line : STRING     DECLARE NewLine : STRING     OPENFILE "Loyalty.txt" FOR READ      WHILE NOT EOF("Loyalty.txt")         READFILE "Loyalty.txt", Line     ENDWHILE     CLOSEFILE "Loyalty.txt"      OPENFILE "Loyalty.txt" FOR APPEND      CustomerID ← STR_TO_NUM(LEFT(Line, 6))      FOR Count ← 1 TO NumToAdd         CustomerID ← CustomerID + 1         OUTPUT CustomerID         NewLine ← NUM_TO_STR(CustomerID) &amp; ",0"         WRITEFILE "Loyalty.txt", NewLine     NEXT Count      CLOSEFILE "Loyalty.txt" ENDPROCEDURE </pre> <p>Mark as follows:</p> <p><b>MP1</b> All variables used are declared using the correct type including a string <b>and</b> an integer</p> <p><b>MP2</b> Open file in read mode <b>and</b> close (before opening in append mode)</p> <p><b>MP3</b> Conditional loop with <u>EOF("Loyalty.txt")</u></p> <p><b>MP4</b> Read <u>Line</u> from file // Count number of records in the file</p> <p><b>MP5</b> Open the file in append mode <b>and</b> subsequently close</p> <p><b>MP6</b> Extract <u>CustomerID</u> from <u>Line</u> <b>and</b> convert to integer // use count from MP4 to generate last <u>CustomerID</u> stored</p> <p><b>MP7</b> A (count controlled) loop for the number of customers to add</p> <p><b>MP8</b> Increment <u>CustomerID</u> <b>and</b> output the new <u>CustomerID</u> in loop</p> <p><b>MP9</b> Create string for new <u>CustomerID</u> <b>and</b> write to file in loop</p> <p><b>Max 8</b></p> |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(b)(ii) | <p><b>MP1</b> Check if the text file <u>loyalty.txt</u> exists / is empty</p> <p><b>MP2</b> If file does not exist it must be created (using write mode)</p> <p><b>MP3</b> If the file has to be created / is empty then set the first <u>CustomerID</u> to 100001</p> <p><b>MP4</b> Write "100001,0" to <u>loyalty.txt</u> (using write mode)</p> <p><b>MP5</b> For all but the first customer (to be added to the empty file) <b>use the existing code / module</b></p> <p><b>Max 4</b></p> |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4(a) | <p><b>MP1</b> <b>Open</b> the file (<u>Sales.txt</u>) in <b>read mode</b> <b>and</b> subsequently close</p> <p><b>MP2</b> <b>Read a line</b> (from the text file)</p> <p><b>MP3</b> Convert the line read (string) to a number</p> <p><b>MP4</b> If the number is greater than 500 <b>and</b> then output week number</p> <p><b>MP5</b> Increment week number (must have been Initialised ...)</p> <p><b>MP6</b> Repeat from step 2 for 52 times // Repeat from step 2 until end of file</p> <p><b>Max 5</b></p>                                                                                                                                                                              |
| 4(b) | <p><b>MP1</b> Going through the program <b>a line / statement at a time</b><br/> // doing a <u>dry run</u><br/> // <u>single-stepping</u> (at run-time with an IDE)<br/> // Peer testing/review is carried out</p> <p><b>MP2</b> Creating a <u>trace table</u></p> <p><b>MP3</b> Draw up <b>test data</b> // draw up list of <b>inputs with expected output</b><br/> // boundary, normal, abnormal data is used</p> <p><b>MP4</b> Errors will be indicated when a <b>variable / output</b> gives an <b>unexpected value</b></p> <p><b>MP5</b> Faults in the logic of the program can be detected // Errors may be indicated by an unexpected path through the program</p> <p><b>Max 3</b></p> |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(a)(i)  | <p><b>Example solution:</b></p> <pre> PROCEDURE UpdateVisit(BYREF LastVisitDate : DATE, __                         BYREF LoyaltyPoints : INTEGER)      IF MONTH(TODAY()) = MONTH(LastVisitDate) THEN         LoyaltyPoints ← LoyaltyPoints + 4     ELSE         LoyaltyPoints ← LoyaltyPoints + 1     ENDIF      LastVisitDate ← TODAY()  ENDPROCEDURE </pre> <p><b>MP1</b> Procedure heading <b>and</b> ending <b>and</b> two parameters of correct type<br/> <b>MP2</b> ... both passed BYREF</p> <p><b>MP3</b> Use of TODAY() function to obtain current date<br/> <b>MP4</b> Use MONTH function (×2) <b>and</b> one parameter is the header parameter <b>and</b> 2nd parameter is TODAY() / assigned variable</p> <p><b>MP5</b> If visit is same month, increase LoyaltyPoints by 4 otherwise increase by 1<br/> <b>MP6</b> LastVisitDate set to TODAY() / 'current date' variable (assigned or unassigned)<br/> <b>Max 5</b></p> |
| 7(a)(ii) | <pre> DAYINDEX(LastVisitDate) = DAYINDEX(TODAY()) </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(b)(i) | <p><b>Example solution:</b></p> <pre> FUNCTION MondayCheck(CustomerID : STRING) RETURNS                                                     INTEGER     DECLARE LoyaltyPoints : INTEGER     DECLARE LoyaltyPointsString : STRING     DECLARE Line : STRING      Line ← FindCustomer(CustomerID)     LoyaltyPointsString ← MID(Line, 8, LENGTH(Line) -                                 16))     LoyaltyPoints ← STR_TO_NUM(LoyaltyPointsString)      IF DAYINDEX(<u>TODAY()</u>) = 2 THEN         LoyaltyPoints ← LoyaltyPoints + 10     ENDIF      RETURN LoyaltyPoints ENDFUNCTION </pre> <p><b>MP1</b> Value is returned from FindCustomer(CustomerID) and stored<br/> <b>MP2</b> Function MID used<br/> <b>MP3</b> Attempt to extract LoyaltyPoints from the string returned by FindCustomer()<br/> <b>MP4</b> Correct extraction of LoyaltyPoints from the string returned by FindCustomer()<br/> <b>MP5</b> Conversion of LoyaltyPoints extracted to an integer<br/> <b>MP6</b> Check - is today's date a Monday using <u>TODAY()</u><br/> <b>MP7</b> ... increase LoyaltyPoints by 10<br/> <b>MP8</b> Return LoyaltyPoints</p> |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7(b)(i)  | <p>Alternative example solution:</p> <p>Uses a loop to calculate the number of digits in the <code>LoyaltyPoints</code> string</p> <pre>FUNCTION MondayCheck(CustomerID : STRING) RETURNS INTEGER      DECLARE LoyaltyPoints : INTEGER     DECLARE LoyaltyPointsString : STRING     DECLARE Index : INTEGER     DECLARE Line : STRING      Line ← FindCustomer(CustomerID)     Index ← 9      WHILE MID(Line, Index, 1) &lt;&gt; ','         Index ← Index + 1     ENDWHILE     // calculate the number of digits in LoyaltyPoints     string     LoyaltyPointsString ← MID(Line, 8, Index - 8)     LoyaltyPoints ← STR_TO_NUM(LoyaltyPointsString)      IF DAYINDEX(TODAY()) = 2 THEN         LoyaltyPoints ← LoyaltyPoints + 10     ENDIF      RETURN LoyaltyPoints  ENDFUNCTION</pre> |
| 7(b)(ii) | <p>Example solution:</p> <pre>DayInt ← STR_TO_NUM(LEFT&gt;LastVisitDateString, 2)) MonthInt ← STR_TO_NUM(MID&gt;LastVisitDateString, 3, 2)) YearInt ← STR_TO_NUM(RIGHT&gt;LastVisitDateString, 4))  LastVisitDate ← SETDATE(DayInt, MonthInt, YearInt)</pre> <p>Mark as follows:</p> <p><b>MP1</b> Use of one substring function</p> <p><b>MP2</b> DayInt, MonthInt and YearInt all correctly extracted</p> <p><b>MP3</b> Use of STR_TO_NUM x 3</p> <p><b>MP4</b> SETDATE() function used to convert to a date type and assigned to LastVisitDate</p>                                                                                                                                                                                                                                    |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12   | <p>One mark for each correctly completed line (Max 5)</p> <pre>DECLARE Location : INTEGER DECLARE Item : STRING DECLARE Continue : BOOLEAN DECLARE Answer : CHAR Continue ← TRUE OPENFILE "StockList.dat" FOR RANDOM WHILE Continue     OUTPUT "Enter a location between 1 and 500: "     INPUT Location     SEEK "StockList.dat", Location     GETRECORD "StockList.dat", Item     IF Item = "" THEN         OUTPUT "This record is missing"     ELSE         OUTPUT "The item in stock is ", Item     ENDIF     OUTPUT "Another location (Y or N)?"     INPUT Answer     IF Answer &lt;&gt; 'Y' THEN         Continue ← FALSE     ENDIF ENDWHILE CLOSEFILE "StockList.dat" OUTPUT "End of program"</pre> |
| 3(a) | <p>1 mark each</p> <ul style="list-style-type: none"> <li>HighScores created as 2D array, (of strings) with (min) 7 × 3 elements (local to main) ...</li> <li>... all elements initialised to empty string ("")</li> </ul> <p>e.g.</p> <p><b>Python</b></p> <pre>HighScores = [] #String, 7 x 3 HighScores = [['' for x in range(3)] for y in range(7)]</pre> <p><b>VB.NET</b></p> <pre>Dim HighScores(7, 3) As String For(X = 0 to 7)     For(Y = 0 to 3)         HighScores(X, Y) = ""     Next Y Next X</pre> <p><b>Java</b></p> <pre>String[][] HighScores = new String[7][3]; for(Int X = 0; X &lt;7; X++){     for(Int Y = 0; Y &lt; 3; Y++){         HighScores[X][Y] = "";     } }</pre>           |
| 3(b) | <p>1 mark each</p> <ul style="list-style-type: none"> <li>Function header (and end where appropriate) that returns populated array</li> <li>Opening text file to read and closing the file in an appropriate place</li> <li>Looping through 7 players/to EOF/21 times ...</li> <li>... reading in each group of 3 data items and storing each in separate element in 2D array for each player</li> <li>Exception handling with all file handling within the try, appropriate catch and an output</li> </ul>                                                                                                                                                                                                |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(b) | <p><b>e.g.</b></p> <p><b>Python</b></p> <pre>def ReadData():     Temp = []     HighScores = []     try:         File = open("HighScoreTable.txt")         Temp = File.read().split("\n")         File.close()     except:         print("No file found")     NumberRecords = len(Temp)-1     Counter = 0      while Counter &lt; NumberRecords:          HighScores.append([Temp[Counter], Temp[Counter+1], Temp[Counter+2]])         Counter = Counter + 3      return HighScores</pre> <p><b>VB.NET</b></p> <pre>Function ReadData()     Dim TextFile As String = "HighScoreTable.txt"      Dim HighScores(7, 3) As String     Try         Dim FileReader As New System.IO.StreamReader(TextFile)         Dim Counter As Integer = 0</pre> |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(b) | <pre>While Counter &lt; 8     HighScores(Counter, 0) = FileReader.ReadLine()     HighScores(Counter, 1) = FileReader.ReadLine()     HighScores(Counter, 2) = FileReader.ReadLine()      Counter = Counter + 1  End While FileReader.Close() Catch ex As Exception     Console.WriteLine("No file found") End Try Return HighScores End Function</pre> <p><b>Java</b></p> <pre>public static String[][] ReadData(){     String TextFile = "HighScoreTable.txt";     String[][] HighScores = new String[7][3];      try{         FileReader f = new FileReader(TextFile);         BufferedReader Reader = new BufferedReader(f);         for(Integer X = 0; X &lt; 7; X++){             try{</pre> |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(b) | <pre>        HighScores[X][0] = Reader.readLine();         HighScores[X][1] = Reader.readLine();         HighScores[X][2] = Reader.readLine();     }catch(IOException ex){}     }     try{         Reader.close();     }catch(IOException ex){}     return HighScores; }catch(FileNotFoundException e){     System.out.println("File not found"); } return HighScores; }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 3(c) | <p>1 mark each</p> <ul style="list-style-type: none"> <li>• Procedure (header and end) taking (min) 1 parameter (2D array), looping through each of the first dimension in array</li> <li>• ... outputting all data in correct format</li> </ul> <p><b>e.g.</b></p> <p><b>Python</b></p> <pre>def OutputHighScores(HighScores):     for x in range(0, len(HighScores)):         print(HighScores[x][0], "reached level", HighScores[x][1], "with a score of",               HighScores[x][2])</pre> <p><b>VB.NET</b></p> <pre>Sub OutputHighScores(HighScores(),)     For x = 0 To 6         Console.WriteLine(HighScores(x, 0) &amp; " reached level " &amp; HighScores(x, 1) &amp; " with a score of " &amp; HighScores(x, 2))     Next End Sub</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 3(c) | <p><b>Java</b></p> <pre>public static void OutputHighScores(String[][] HighScores){     for(Integer x = 0; x &lt; 7; x++){         System.out.println(HighScores[x][0] + " reached level " + HighScores[x][1] + " with a score of " + HighScores[x][2]);     } }</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3(d) | <p>1 mark each</p> <ul style="list-style-type: none"> <li>• Function header (and end taking array as parameter) returning sorted array.</li> <li>• Comparing the levels and swapping <b>all</b> dimensions when in incorrect order</li> <li>• Comparing scores when levels are the same and swapping <b>all</b> dimensions when in incorrect order</li> <li>• Correct loops and comparisons to put data in correct order</li> </ul> <p><b>e.g.</b></p> <p><b>Python</b></p> <pre>def SortScores(HighScores):     Counter = 0     ArrayLength = len(HighScores)      for x in range(ArrayLength-1):         for y in range(0, ArrayLength-x-1):             if int(HighScores[y][1]) &lt; int(HighScores[y + 1][1]):                 HighScores[y], HighScores[y + 1] = HighScores[y + 1], HighScores[y]             elif int(HighScores[y][1]) == int(HighScores[y+1][1]):                 if int(HighScores[y][2]) &lt; int(HighScores[y+1][2]):                     HighScores[y], HighScores[y + 1] = HighScores[y + 1], HighScores[y]     return HighScores</pre> <p><b>VB.NET</b></p> <pre>Function SortScores(HighScores)     Dim ArrayLength As Integer = 6     Dim Temp1 As String</pre> |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(d) | <pre> Dim Temp2 As String Dim Temp3 As String For x = 0 To ArrayLength - 1     For y = 0 To ArrayLength - x - 1         If Integer.Parse(HighScores(y, 1)) &lt; Integer.Parse(HighScores(y + 1, 1))             Then                 Temp1 = HighScores(y, 0)                 Temp2 = HighScores(y, 1)                 Temp3 = HighScores(y, 2)                 HighScores(y, 0) = HighScores(y + 1, 0)                 HighScores(y, 1) = HighScores(y + 1, 1)                 HighScores(y, 2) = HighScores(y + 1, 2)                 HighScores(y + 1, 0) = Temp1                 HighScores(y + 1, 1) = Temp2                 HighScores(y + 1, 2) = Temp3             ElseIf Integer.Parse(HighScores(y, 1)) = Integer.Parse(HighScores(y + 1, 1)) Then                 If Int(HighScores(y, 2)) &lt; Int(HighScores(y + 1, 2)) Then                     Temp1 = HighScores(y, 0)                     Temp2 = HighScores(y, 1)                     Temp3 = HighScores(y, 2)                     HighScores(y, 0) = HighScores(y + 1, 0)                     HighScores(y, 1) = HighScores(y + 1, 1)                     HighScores(y, 2) = HighScores(y + 1, 2)                     HighScores(y + 1, 0) = Temp1                     HighScores(y + 1, 1) = Temp2                     HighScores(y + 1, 2) = Temp3                 End If             End If         Next y     Next x     Return HighScores End Function </pre> <p>Java</p> <pre> public static String[][] SortScores(String[][] HighScores){ </pre> |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(d) | <pre> Integer ArrayLength = 6; String Temp1; String Temp2; String Temp3;  for(Integer x = 0; x &lt; ArrayLength; x++){     for(Integer y = 0; y &lt; ArrayLength - x; y++){         if(Integer.parseInt(HighScores[y][1]) &lt; Integer.parseInt(HighScores[y+1][1])){             Temp1 = HighScores[y][0];             Temp2 = HighScores[y][1];             Temp3 = HighScores[y][2];             HighScores[y][0] = HighScores[y+1][0];             HighScores[y][1] = HighScores[y+1][1];             HighScores[y][2] = HighScores[y+1][2];             HighScores[y+1][0] = Temp1;             HighScores[y+1][1] = Temp2;             HighScores[y+1][2] = Temp3;         }else if(Integer.parseInt(HighScores[y][1]) == Integer.parseInt(HighScores[y+1][1])){             if(Integer.parseInt(HighScores[y][2]) &lt; Integer.parseInt(HighScores[y+1][2])){                 Temp1 = HighScores[y][0];                 Temp2 = HighScores[y][1];                 Temp3 = HighScores[y][2];                 HighScores[y][0] = HighScores[y+1][0];                 HighScores[y][1] = HighScores[y+1][1];                 HighScores[y][2] = HighScores[y+1][2];                 HighScores[y+1][0] = Temp1;                 HighScores[y+1][1] = Temp2;                 HighScores[y+1][2] = Temp3;             }         }     } } </pre> |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3(d)     | <pre>     }     }     return HighScores; } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 3(e)(i)  | <p>1 mark each</p> <ul style="list-style-type: none"> <li>Code statements in order:<br/> HighScores = ReadData()<br/> HighScores = SortScores(HighScores) // HighScores = SortScores()</li> <li>Code statements in order:<br/> OUTPUT "Before"<br/> OutputHighScores(HighScores) unsorted<br/> OUTPUT "After"<br/> OutputHighScores(HighScores) sorted</li> </ul> <p>e.g.</p> <p>Python</p> <pre> HighScores = [] HighScores = ReadData() print("Before") OutputHighScores(HighScores) HighScores = SortScores(HighScores) print("After") OutputHighScores(HighScores) </pre> <p>VB.NET</p> <pre> Sub Main(args As String())     Dim HighScores(7, 3) As String     HighScores = ReadData()     Console.WriteLine("Before")     OutputHighScores(HighScores)     HighScores = SortScores(HighScores)     Console.WriteLine("After")     OutputHighScores(HighScores) End Sub </pre> |
| 3(e)(i)  | <p>Java</p> <pre> public static void main(String args[]){     String[][] HighScores = new String[7][3];     HighScores = ReadData();     System.out.println("Before");     OutputHighScores(HighScores);     HighScores = SortScores(HighScores);     System.out.println("After");     OutputHighScores(HighScores); } </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 3(e)(ii) | <p>Output showing 'Before' and players and scores in correct format before sorting<br/> Output showing 'After' players and scores in correct order and format after sorting</p> <p>e.g.</p> <pre> Before GHEH got to level 3 with a score of 10 RWQW got to level 4 with a score of 20 MMND got to level 4 with a score of 18 RFOO got to level 5 with a score of 20 XXHD got to level 3 with a score of 19 QWSD got to level 3 with a score of 15 JGHE got to level 5 with a score of 22 After JGHE got to level 5 with a score of 22 RFOO got to level 5 with a score of 20 RWQW got to level 4 with a score of 20 MMND got to level 4 with a score of 18 XXHD got to level 3 with a score of 19 QWSD got to level 3 with a score of 15 GHEH got to level 3 with a score of 10 </pre>                                                                                             |