CAMBRIDGE
International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/43**

Paper 4 Practical **October/November 2025**

MARK SCHEME

Maximum Mark: 75

Published

This document consists of **39** printed pages.

Generic Marking Principles

These general marking principles must be applied by all examiners when marking candidate answers. They should be applied alongside the specific content of the mark scheme or generic level descriptions for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

- the specific content of the mark scheme or the generic level descriptors for the question
- the specific skills defined in the mark scheme or in the generic level descriptors for the question
- the standard of response required by a candidate as exemplified by the standardisation scripts.

GENERIC MARKING PRINCIPLE 2:

Marks awarded are always **whole marks** (not half marks, or other fractions).

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

- marks are awarded for correct/valid answers, as defined in the mark scheme. However, credit is given for valid answers which go beyond the scope of the syllabus and mark scheme, referring to your Team Leader as appropriate
- marks are awarded when candidates clearly demonstrate what they know and can do
- marks are not deducted for errors
- marks are not deducted for omissions
- answers should only be judged on the quality of spelling, punctuation and grammar when these features are specifically assessed by the question as indicated by the mark scheme. The meaning, however, should be unambiguous.

GENERIC MARKING PRINCIPLE 4:

Rules must be applied consistently, e.g. in situations where candidates have not followed instructions or in the application of generic level descriptors.

GENERIC MARKING PRINCIPLE 5: Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).
GENERIC MARKING PRINCIPLE 6: Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Annotations guidance for centres

Examiners use a system of annotations as a shorthand for communicating their marking decisions to one another. Examiners are trained during the standardisation process on how and when to use annotations. The purpose of annotations is to inform the standardisation and monitoring processes and guide the supervising examiners when they are checking the work of examiners within their team. The meaning of annotations and how they are used is specific to each component and is understood by all examiners who mark the component.

We publish annotations in our mark schemes to help centres understand the annotations they may see on copies of scripts. Note that there may not be a direct correlation between the number of annotations on a script and the mark awarded. Similarly, the use of an annotation may not be an indication of the quality of the response.

The annotations listed below were available to examiners marking this component in this series.

Annotations

Annotation	**Meaning**
BOD	Benefit of the doubt
λ	To indicate where a key word/phrase/code is missing
✗	Incorrect
FT	Follow through
〰	Indicate a point in an answer
Highlighted text	To draw attention to a particular aspect or to indicate where parts of an answer have been combined
I	Ignore
NAQ	Not answered question
NE	No examples or not enough

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

Mark scheme abbreviations

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

Question	Answer	Marks

1(a)(i)	1 mark each
	• Class header (and end where appropriate)
	• Declaring `Code` as string and `Value` as integer
	• Constructor header (and end where appropriate) within class …
	• … taking two parameters …
	• … assigning parameters to attributes
	
	Example program code.
	Java
	`class BoardObject{`
	`      public String Code;`
	`      public Integer Value;`
	`      public BoardObject(String pCode, Integer pValue){`
	`            Code = pCode;`
	`            Value = pValue;`
	`      }`
	`}`
	
	VB.NET
	`Public Class BoardObject`
	`     Private Code As String`
	`     Private Value As Integer`
	`     Sub New(pCode, pValue)`
	`         Code = pCode`
	`         Value = pValue`
	`     End Sub`
	`End Class`
	
	Python
	`class BoardObject():`
	`    def __init__(self, Code, Value):`
	`        self.Code = Code #string`
	`        self.Value = Value # integer`
		5

Question	Answer	Marks

1(a)(ii)	1 mark each
	• 1 get header (and end where appropriate) with no parameter …
	• … returning correct value without overriding
	• 2nd correct get method
	
	Example program code
	
	Java
	`public String GetCode(){`
	`      return Code;`
	`}`
	`public Integer GetValue(){`
	`      return Value;`
	`}`
	
	VB.NET
	`Function GetCode()`
	`    Return Code`
	`End Function`
	
	`Function GetValue()`
	`    Return Value`
	`End Function`
	
	Python
	`def GetCode(self):`
	`    return self.Code`
	
	`def GetValue(self):`
	`    return self.Value`
		3

Question	Answer	Marks

1(a)(iii)	1 mark each
	• Creating one instance of `BoardObject` **and** storing in correct variable …
	• … with correct parameters
	• Remaining four created correctly and stored
	
	Example program code
	
	Java
	`BoardObject Object1 = new BoardObject("A",2);`
	`BoardObject Object2 = new BoardObject("B",3);`
	`BoardObject Object3 = new BoardObject("C",5);`
	`BoardObject Object4 = new BoardObject("D",2);`
	`BoardObject Object5 = new BoardObject("E",7);`
	
	VB.NET
	`Dim Object1 As BoardObject = New BoardObject("A", 2)`
	`Dim Object2 As BoardObject = New BoardObject("B", 3)`
	`Dim Object3 As BoardObject = New BoardObject("C", 5)`
	`Dim Object4 As BoardObject = New BoardObject("D", 2)`
	`Dim Object5 As BoardObject = New BoardObject("E", 7)`
	
	Python
	`Object1 = BoardObject("A",2)`
	`Object2 = BoardObject("B",3)`
	`Object3 = BoardObject("C",5)`
	`Object4 = BoardObject("D",2)`
	`Object5 = BoardObject("E",7)`
		3

Question	Answer	Marks

1(b)(i)	1 mark each
	• Class header (and end where appropriate)
	• Constructor header (and end where appropriate) within class
	• Declaration of 2D array with 10 × 10 elements of type `BoardObject`
	• Storing `BoardObject` object with `Code` `"-"` and `Value` `0` in each array element
	
	Example program code
	
	Java
	`class Board{`
	`      private BoardObject[][] TheBoard = new BoardObject[10][10];`
	`      public Board(){`
	`            for(Integer x = 0; x < 10; x++){`
	`                  for(Integer y = 0; y < 10; y++){`
	`                        TheBoard[x][y] = new BoardObject("-",0);`
	`                  }}}}`
	VB.NET
	`Public Class Board`
	`    Private TheBoard(9, 9) As BoardObject`
	`    Sub New()`
	`        For x = 0 To 9`
	`            For y = 0 To 9`
	`                TheBoard(x, y) = New BoardObject("-", 0)`
	`            Next`
	`        Next`
	`    End Sub`
	
	`End Class`
		4

Question	Answer	Marks

1(b)(i)	Python
	`class Board():`
	`    def __init__(self):`
	`        self.TheBoard = [] #type BoardObject`
	`        for x in range(10):`
	`            TempList = []`
	`            for y in range(10):`
	`                TempList.append(BoardObject("-",0))`
	`            self.TheBoard.append(TempList)`	

1(b)(ii)	1 mark each
	• Get method header taking 2 (integer) parameters …
	• … returning the `BoardObject` at `Board` position of parameters
	
	Example program code
	Java
	`public BoardObject GetObject(Integer Rowpos, Integer Columnpos){`
	`           return TheBoard[Rowpos][Columnpos];`
	`}`
	VB.NET
	`Function GetObject(Rowpos, Columnpos)`
	`     Return TheBoard(Rowpos, Columnpos)`
	` End Function`
	
	Python
	`def GetObject(self, Rowpos, Columnpos):`
	`        return self.TheBoard[Rowpos][Columnpos]`
		2

Question	Answer	Marks

1(b)(iii)	1 mark each
	• Set method header taking three parameters (`TheObject`, row, column) …
	• … storing parameter `TheObject` in `TheBoard` at parameters row and column
	
	Example program code
	Java
	`public void SetObject(BoardObject TheObject, Integer Rowpos, Integer Columnpos){`
	`      TheBoard[Rowpos][Columnpos] = TheObject;`
	`}`
	VB.NET
	`   Sub SetObject(TheObject, Rowpos, Columnpos)`
	`    TheBoard(Rowpos, Columnpos) = TheObject`
	`End Sub`
	
	Python
	`def SetObject(self, TheObject, Rowpos, Columnpos):`
	`        self.TheBoard[Rowpos][Columnpos] = TheObject`
		2

Question	Answer	Marks

1(b)(iv)	1 mark each
	• Method `DisplayBoard()` header (and end where appropriate) **and** using `GetCode()`
	• Outputting `Code` of all `BoardObject` elements in both indices (10 × 10)
	• … with each row on one line and space between each value
	
	Example program code
	
	Java
	`public void DisplayBoard(){`
	`            String OutputLine;`
	`            for(Integer x = 0; x < 10; x++){`
	`                  OutputLine = "";`
	`                  for(Integer y = 0; y < 10; y++){`
	`                        OutputLine = OutputLine + TheBoard[x][y].GetCode() + " ";`
	`                  }`
	`                  System.out.println(OutputLine);`
	`            }`
	`      }`
	
	VB.NET
	`Sub DisplayBoard()`
	`    Dim OutputLine As String`
	`    For x = 0 To 9`
	`        OutputLine = ""`
	`        For y = 0 To 9`
	`            OutputLine = OutputLine & TheBoard(x, y).GetCode() & " "`
	`        Next`
	`        Console.WriteLine(OutputLine)`
	`    Next`
	`End Sub`
		3

Question	Answer	Marks

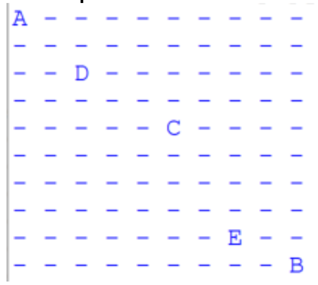
1(b)(iv)	Python
	`def DisplayBoard(self):`
	
	`        for x in range(10):`
	`            OutputLine = ""`
	`            for y in range(10):`
	`                OutputLine = OutputLine + str(self.TheBoard[x][y].GetCode()) + " "`
	`            print(OutputLine)`	

Question	Answer	Marks

1(c)(i)	1 mark each
	• Creating instance of `Board` and storing it
	• Storing all 5 objects in correct positions
	• Calling `DisplayBoard()`
	
	Example program code
	
	Java
	`Board GameBoard =new Board();`
	`BoardObject Object1 = new BoardObject("A",2);`
	`BoardObject Object2 = new BoardObject("B",3);`
	`BoardObject Object3 = new BoardObject("C",5);`
	`BoardObject Object4 = new BoardObject("D",2);`
	`BoardObject Object5 = new BoardObject("E",7);`
	
	`GameBoard.SetObject(Object1, 0, 0);`
	`GameBoard.SetObject(Object2, 9, 9);`
	`GameBoard.SetObject(Object3, 4, 5);`
	`GameBoard.SetObject(Object4, 2, 2);`
	`GameBoard.SetObject(Object5, 8, 7);`
	`GameBoard.DisplayBoard();`
	
	VB.NET
	`Dim GameBoard As Board = New Board()`
	` Dim Object1 As BoardObject = New BoardObject("A", 2)`
	` Dim Object2 As BoardObject = New BoardObject("B", 3)`
	` Dim Object3 As BoardObject = New BoardObject("C", 5)`
	` Dim Object4 As BoardObject = New BoardObject("D", 2)`
	` Dim Object5 As BoardObject = New BoardObject("E", 7)`
	
	` GameBoard.SetObject(Object1, 0, 0)`
	` GameBoard.SetObject(Object2, 9, 9)`
	` GameBoard.SetObject(Object3, 4, 5)`
	` GameBoard.SetObject(Object4, 2, 2)`
	` GameBoard.SetObject(Object5, 8, 7)`
	` GameBoard.DisplayBoard()`
		3

Question	Answer	Marks

1(c)(i)	Python
	`GameBoard = Board()`
	`Object1 = BoardObject("A",2)`
	`Object2 = BoardObject("B",3)`
	`Object3 = BoardObject("C",5)`
	`Object4 = BoardObject("D",2)`
	`Object5 = BoardObject("E",7)`
	`GameBoard.SetObject(Object1, 0, 0)`
	`GameBoard.SetObject(Object2, 9, 9)`
	`GameBoard.SetObject(Object3, 4, 5)`
	`GameBoard.SetObject(Object4, 2, 2)`
	`GameBoard.SetObject(Object5, 8, 7)`
	`GameBoard.DisplayBoard()`	

1(c)(ii)	1 mark for screenshot showing board contents
	
	Example
	
	`A - - - - - - - - -`
	`- - - - - - - - - -`
	`- - D - - - - - - -`
	`- - - - - - - - - -`
	`- - - - - C - - - -`
	`- - - - - - - - - -`
	`- - - - - - - - - -`
	`- - - - - - - - - -`
	`- - - - - - - E - -`
	`- - - - - - - - - B`
		1

Question	Answer	Marks

1(d)(i)	1 mark each to max 4
	• Taking x-axis and y-axis as input repeatedly until each value is between 0 and 9 (inclusive)
	• Calling `Board.GetObject()` with input x-axis and y-axis values …
	• … checking if object exists e.g. if `Code` is `"-"`
	• … outputting "Miss" if no `BoardObject` **and** outputting `Code` and `Value` in an appropriate message if there is a `BoardObject`
	
	Example program code
	
	Java
	`Scanner scanner = new Scanner(System.in);`
	`      Integer InputRow = -1;`
	`      while(InputRow < 0 || InputRow > 9){`
	`            System.out.println("Enter the row position between 0 and 9 ");`
	`            InputRow = Integer.parseInt(scanner.nextLine());`
	`      }`
	`      Integer InputColumn = -1;`
	`      while(InputColumn < 0 || InputColumn > 9){`
	`            System.out.println("Enter the column position between 0 and 9 ");`
	`               InputColumn = Integer.parseInt(scanner.nextLine());`
	`      }`
	`      BoardObject GuessObject = GameBoard.GetObject(InputRow, InputColumn);`
	`      if((GuessObject.GetCode()).equals("-")){`
	`          System.out.println("Miss");`
	`      }else{`
	`          System.out.println("You found " + GuessObject.GetCode() + " with value " +`
	`GuessObject.GetValue());`
	`      }`
	VB.NET
	`Dim InputRow, InputColumn As Integer`
	`InputRow = -1`
	`InputColumn = -1`
	`While InputRow < 0 Or InputRow > 9`
	`    Console.WriteLine("Enter the row position between 0 and 9 ")`
	`    InputRow = Console.ReadLine()`
	`End While`
		4

Question	Answer	Marks

1(d)(i)	
	```
	While InputColumn < 0 Or InputColumn > 9
	    Console.WriteLine("Enter the column position between 0 and 9 ")
	    InputColumn = Console.ReadLine()
	End While

	Dim GuessObject As BoardObject = GameBoard.GetObject(InputRow, InputColumn)
	If GuessObject.GetCode() = "-" Then
	    Console.WriteLine("Miss")
	Else
	    Console.WriteLine("You found " & GuessObject.GetCode() & " with value " &
	GuessObject.GetValue())
	End If
	```
	Python
	```
	InputRow = -1
	while InputRow < 0 or InputRow > 9:
	    InputRow = int(input("Enter the row position between 0 and 9 "))

	InputColumn = -1
	while InputColumn < 0 or InputColumn > 9:
	    InputColumn = int(input("Enter the column position between 0 and 9 "))

	GuessObject = GameBoard.GetObject(InputRow, InputColumn)
	if GuessObject.GetCode() == "-":
	    print("Miss")
	else:
	    print("You found " + str(GuessObject.GetCode()) + " with value " +
	str(GuessObject.GetValue()))
	```

Question	Answer	Marks

1(d)(ii)	1 mark for screenshot showing inputs and correct output
	Row 10 4
	Column -1 5
	Output C 5
	
	Example
	`Enter the row position between 0 and 9 10`
	`Enter the row position between 0 and 9 4`
	`Enter the column position between 0 and 9 -1`
	`Enter the column position between 0 and 9 5`
	`You found C with value 5`
		1

Question	Answer	Marks

2(a)	1 mark each
	• `Queue` declared as a (global) 1D array of 100 string elements all initialised to `""`
	• `QueueHead` and `QueueTail` initialised with `-1`, `NumberItems` initialised to `0`
	
	Example program code
	Java
	`public static String[] Queue = new String[100];`
	`public static Integer QueueHead;`
	`public static Integer QueueTail;`
	`public static Integer NumberItems;`
	`for(int x = 0; x < 100; x++){`
	`      Queue[x] = "";`
	`}`
	`QueueHead = -1;`
	`QueueTail = -1;`
	`NumberItems = 0;`
	
	VB.NET
	`    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`
		2

Question	Answer	Marks

2(a)	Python
	`global Queue, QueueHead, QueueTail, NumberItems`
	`Queue = []`
	`    for x in range(100):`
	`        Queue.append("")`
	`    QueueHead = -1`
	`    QueueTail = -1`
	`    NumberItems = 0`	

Question	Answer	Marks

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){`
	        `Queue[QueueTail+1] = TheData;`
	        `QueueTail++;`
	        `NumberItems++;`
	        `return true;`
	    `}else{`
	        `return false;`
	    `}`
	
	`}`
		5

Question	Answer	Marks

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`
	`        Queue(QueueTail + 1) = TheData`
	`        QueueTail += 1`
	`        NumberItems += 1`
	`        Return True`
	`    Else`
	`        Return False`
	`    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:`
	`        Queue[QueueTail+1] = TheData`
	`        QueueTail +=1`
	`        NumberItems +=1`
	`        return True`
	`    else:`
	`        return False`	

Question	Answer	Marks

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`
		3

Question	Answer	Marks

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`	

Question	Answer	Marks

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");`
	`        }`
	`    }`
		5

Question	Answer	Marks

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()`	

Question	Answer	Marks

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;`
	`          }}`
		6

Question	Answer	Marks

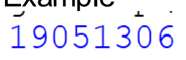
2(e)	VB.NET
	`Sub Compress()`
	`     Dim First As String = Dequeue()`
	`     Dim NewLine As String = ""`
	`     Dim Count As Integer`
	`     Dim NextChar As String`
	`     While NumberItems > 0 And First <> "False"`
	`         Count = 1`
	`         NextChar = Dequeue()`
	
	`         While NextChar =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

Question	Answer	Marks

2(f)(i)	1 mark each
	• Calling `ReadData()` then `Compress()`
	• Outputting compressed string
	
	Example program code
	
	Java
	`public static void main(String args[]){`
	`            NewString = "";`
	`            for(int x = 0; x < 100; x++){`
	`                      Queue[x] = "";`
	`                         }`
	`                       QueueHead = -1;`
	`                       QueueTail = -1;`
	`                     NumberItems = 0;`
	
	`            ReadData();`
	`            Compress();`
	`            System.out.println(NewString);`
	`      }`
	
	VB.NET
	`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`
		2

Question	Answer	Marks

2(f)(i)	Python
	
	`NewString = ""`
	`Queue = []`
	`for x in range(100):`
	`        Queue.append("")`
	`QueueHead = -1`
	`QueueTail = -1`
	`NumberItems = 0`
	`ReadData()`
	`Compress()`
	`print(NewString)`	

2(f)(ii)	1 mark for screenshot showing output Example  `19051306`	**1**

Question	Answer	Marks

3(a)(i)	1 mark each
	• Function header (and end) taking three parameters
	• Checking number of elements for 0 and returning 0 (e.g. `ArrayCopy = []` // `len(ArrayCopy) == 0`)
	• (otherwise) comparing first array element to parameter `DataToFind` …
	• … if match return recursive call adding 1
	• … if no match return recursive call without adding 1
	• … all recursive calls have array without first element, `NumberElements-1` and `DataToFind`
	
	Example program code
	
	Java
	`public static Integer RecursiveCount(Integer DataToFind, Integer[] ArrayCopy, Integer NumberElements){`
	`            if(NumberElements > 0){`
	`                  Integer[] NewArray = new Integer[NumberElements-1];`
	`                  for(Integer x = 1; x < NumberElements; x++){`
	`                        NewArray[x-1]= ArrayCopy[x];`
	`                  }`
	
	`                  if(ArrayCopy[0] == DataToFind){`
	`                        return 1 + RecursiveCount(DataToFind, NewArray, NumberElements - 1);`
	`                  }else{`
	`                        return RecursiveCount(DataToFind, NewArray, NumberElements - 1);`
	`                  }`
	`            }else{`
	`                  return 0;`
	`            }`
	`      }`
		6

Question	Answer	Marks

3(a)(i)	VB.NET
	`Function RecursiveCount(DataToFind As Integer, ArrayCopy() As Integer, NumberElements As Integer)`
	
	`    If NumberElements > 0 Then`
	`        Dim NewArray(NumberElements - 1) As Integer`
	`        For x = 1 To NumberElements - 1`
	`            NewArray(x - 1) = ArrayCopy(x)`
	`        Next x`
	
	`        If ArrayCopy(0) = DataToFind Then`
	`            Return 1 + RecursiveCount(DataToFind, NewArray, NumberElements - 1)`
	`        Else`
	`            Return RecursiveCount(DataToFind, NewArray, NumberElements - 1)`
	
	`        End If`
	`    Else`
	`        Return 0`
	
	`    End If`
	`End Function`
	
	Python
	`def RecursiveCount(DataToFind, ArrayCopy, NumberElements):`
	`    if NumberElements > 0:`
	`        NewArray = ArrayCopy[1:]`
	`        if ArrayCopy[0] == DataToFind:`
	`            return 1 + RecursiveCount(DataToFind, NewArray, NumberElements - 1)`
	`        else:`
	`            return RecursiveCount(DataToFind, NewArray, NumberElements - 1)`
	`    else:`
	`        return 0`	

Question	Answer	Marks

3(a)(ii)	1 mark each
	• Storing the correct data in an array
	• Calling `RecursiveCount()` with the correct parameters
	• Outputting the return value
	
	Example program code
	
	Java
	`Integer[] MyArray = new Integer[]{0, 5, 1, 2, 5, 9, 9, 6, 5, 0};`
	`System.out.println(RecursiveCount(0, MyArray, 10));`
	
	VB.NET
	`Dim MyArray() As Integer = {0, 5, 1, 2, 5, 9, 9, 6, 5, 0}`
	
	`Console.WriteLine(RecursiveCount(0, MyArray, 10))`
	
	Python
	`MyArray = [0,5,1,2,5,9,9,6,5,0]`
	`print(RecursiveCount(0, MyArray, 10))`
		3

3(a)(iii)	1 mark for screenshot showing correct output of 2 Example `2`	**1**

3(b)(i)	1 mark for storing the string in a variable
	
	Example program code
	
	Java
	`String Code = "x=0;y=1;x=x+y;y++;";`
	VB.NET
	`    Dim Code As String = "x=0;y=1;x=x+y;y++;"`
	Python
	`Code = "x=0;y=1;x=x+y;y++;"`
		1

Question	Answer	Marks

3(b)(ii)	1 mark each
	• Function header, taking one string parameter
	• Loop e.g. four times/through each character in parameter
	• Comparing character from parameter to `';'` …
	• … concatenate current character to a string until `';'` is found
	• … when `';'` found storing string in array
	• Returning string array without semicolons
	
	Example program code
	Java
	`public static String[] SplitData(String DataString){`
	`        String[] SplitDataArray = new String[10];`
	`        Integer Count = 0;`
	`        String TempString = "";`
	`        String Character = "";`
	`        Integer LastElement = 0;`
	
	`        for(Integer x = 0; x < 4; x++){`
	`            TempString = "";`
	
	`            try{`
	`                Character = String.valueOf(DataString.charAt(Count));`
	`                while(Character.equals(";") == false){`
	`                    TempString = TempString + Character;`
	`                    Count++;`
	`                    Character = String.valueOf(DataString.charAt(Count));`
	`                }`
	
	`                SplitDataArray[LastElement] = TempString;`
	`                LastElement++;`
	
	`            }finally{}`
	`            Count++;`
	`        }`
	`        return SplitDataArray;`
	`    }`
		6

Question	Answer	Marks

3(b)(ii)	VB.NET
	`Function SplitData(DataString)`
	`     Dim SplitDataArray(10) As String`
	`     Dim Count As Integer = 0`
	`     Dim TempString As String = ""`
	`     Dim Character As String = ""`
	`     Dim LastElement As Integer = 0`
	
	`     For x = 0 To 3`
	`         TempString = ""`
	`         Try`
	`             Character = DataString(Count)`
	
	`             While Character <> ";"`
	`                 TempString = TempString + Character`
	`                 Count = Count + 1`
	`                 Character = DataString(Count)`
	
	`             End While`
	
	`             SplitDataArray(LastElement) = TempString`
	`             LastElement = LastElement + 1`
	`         Catch`
	`             Console.WriteLine("No more character")`
	`         End Try`
	`         Count = Count + 1`
	`     Next`
	
	`     Return SplitDataArray`
	` End Function`	

Question	Answer	Marks

3(b)(ii)	Python
	`def SplitData(DataString):`
	`    SplitDataArray = []`
	`    Count = 0`
	`    for x in range(4):`
	`        TempString = ""`
	`        try:`
	`            Character = DataString[Count]`
	`            while Character != ";":`
	`                TempString = TempString + (Character)`
	`                Count += 1`
	`                Character = DataString[Count]`
	`            SplitDataArray.append(TempString)`
	`        except:`
	`            print("No more characters")`
	`        Count += 1`
	`    return SplitDataArray`	

Question	Answer	Marks
3(b)(iii)	1 mark each • Calling `SplitData()` with array as argument **and** storing/using return value • Outputting each element of the returned array on a new line	**2**

Example program code

Java
```
String Code = "x=0;y=1;x=x+y;y++;";
String[] SplitDataArray = new String[10];
SplitDataArray = SplitData(Code);

for(Integer x = 0; x < 4; x++){
    System.out.println(SplitDataArray[x]);
}
```

VB.NET
```
Dim Code As String = "x=0;y=1;x=x+y;y++;"
 Dim SplitDataArray() As String = SplitData(Code)

 For x = 0 To 3
     Console.WriteLine(SplitDataArray(x))
 Next
```

Python
```
Code = "x=0;y=1;x=x+y;y++;"
SplitDataArray = SplitData(Code)
for x in range(4):
    print(SplitDataArray[x])
```

Question	Answer	Marks

3(b)(iv)	1 mark for output
	x=0
	y=1
	x=x+y
	y++
	
	Example
	`x=0`
	`y=1`
	`x=x+y`
	`y++`
		1