Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/42**

Paper 4 Practical **October/November 2022**

MARK SCHEME

Maximum Mark: 75

Published

This mark scheme is published as an aid to teachers and candidates, to indicate the requirements of the examination. It shows the basis on which Examiners were instructed to award marks. It does not indicate the details of the discussions that took place at an Examiners' meeting before marking began, which would have considered the acceptability of alternative answers.

Mark schemes should be read in conjunction with the question paper and the Principal Examiner Report for Teachers.

Cambridge International will not enter into discussions about these mark schemes.

Cambridge International is publishing the mark schemes for the October/November 2022 series for most Cambridge IGCSE™, Cambridge International A and AS Level components and some Cambridge O Level components.

This document consists of **24** 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 descriptors for a question. Each question paper and mark scheme will also comply with these marking principles.

GENERIC MARKING PRINCIPLE 1:

Marks must be awarded in line with:

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

GENERIC MARKING PRINCIPLE 2:

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

GENERIC MARKING PRINCIPLE 3:

Marks must be awarded **positively**:

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

GENERIC MARKING PRINCIPLE 4:

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

GENERIC MARKING PRINCIPLE 5:

Marks should be awarded using the full range of marks defined in the mark scheme for the question (however; the use of the full mark range may be limited according to the quality of the candidate responses seen).

GENERIC MARKING PRINCIPLE 6:

Marks awarded are based solely on the requirements as defined in the mark scheme. Marks should not be awarded with grade thresholds or grade descriptors in mind.

Question	Answer	Marks

1(a)	1 mark per point:
	
	• (global) 2-D array `Jobs` with correct identifier (and Integer data type)
	• … with 100 elements by 2 elements
	• (global) `NumberOfJobs` declared as variable (as Integer)
	
	Example program code:
	
	Java
	`public static Integer[][] Jobs = new Integer[100][2];`
	`public static Integer NumberOfJobs;`
	
	Python
	`Jobs # global integer, 100 by 2 elements`
	`NumberOfJobs # global integer`
	
	VB.NET
	`Dim Jobs(99, 1) As Integer`
	`Dim NumberOfJobs As Integer`
		3

Question	Answer	Marks

1(b)	1 mark per point:
	
	• procedure heading (and end where appropriate) **and** assigns 0 to `NumberOfJobs`
	• **looping** through **both** array element dimensions
	• … assigns –1 to **all** elements
	
	Example program code:
	
	Java
	`public static void Initialise(){`
	`  for(Integer x = 0; x<100;x++){`
	`  for(Integer y = 0; y<2; y++){`
	`  Jobs[x][y] = -1;`
	`  }`
	`  }`
	`  NumberOfJobs = 0;`
	`}`
	
	Python
	`def Initialise():`
	`  global Jobs`
	`  global NumberOfJobs`
	`  for x in range(0, 100):`
	`  Jobs.append([-1,-1])`
	`  NumberOfJobs = 0`
	
	VB.NET
	`Sub Initialise()`
	`  For X = 0 To 99`
	`  For Y = 0 To 1`
	`  Jobs(X, Y) = -1`
	`  Next`
	`  Next`
	`  NumberOfJobs = 0`
	`End Sub`
		3

Question	Answer	Marks

1(c)	1 mark per point (**Max 5**):
	
	• Function header (and end where appropriate) with **two** (integer) parameters
	• Checks if array is full …
	• … if full outputs "Not added"
	• Storing **parameters** job number and priority to **only** the next available array position
	• Incrementing `NumberOfJobs`
	• Outputting "Added" **if successful**
	
	Example program code:
	
	Java
	`public static void AddJob(Integer Description, Integer Priority){`
	`  if(NumberOfJobs == 100){`
	`  System.out.println("Not added");`
	`  }else{`
	`  Jobs[NumberOfJobs][0] = Description;`
	`  Jobs[NumberOfJobs][1] = Priority;`
	`  NumberOfJobs = NumberOfJobs + 1;`
	`  System.out.println("Added");`
	`  }`
	`}`
	
	Python
	`def AddJob(JobNumber, Priority):`
	`  global NumberOfJobs`
	`  global Jobs`
	`  if NumberOfJobs == 100:`
	`  print("Not added")`
	`  else:`
	`  Jobs[NumberOfJobs] = [JobNumber, Priority]`
	`  print("Added")`
	`  NumberOfJobs = NumberOfJobs + 1`
	
	VB.NET
	`Sub AddJob(JobNumber, Priority)`
	`  If NumberOfJobs = 100 Then`
	`  Console.WriteLine("Not added")`
	`  Else`
	`  Jobs(NumberOfJobs, 0) = JobNumber`
	`  Jobs(NumberOfJobs, 1) = Priority`
	`  NumberOfJobs = NumberOfJobs + 1`
	`  Console.WriteLine("Added")`
	`  End If`
	`End Sub`
		5

Question	Answer	Marks

1(d)	1 mark per point:
	
	• Calls `Initialise()` (in the main program)
	• 5 `AddJob` calls with correct values as parameters in correct order
	
	Example program code:
	
	Java
	`public static void main(String args[]){`
	`  Initialise();`
	`  AddJob(12, 10);`
	`  AddJob(526, 9);`
	`  AddJob(33,8);`
	`  AddJob(12,9);`
	`  AddJob(78,1);`
	`}`
	Python
	`Initialise()`
	`AddJob(12,10)`
	`AddJob(526,9)`
	`AddJob(33,8)`
	`AddJob(12,9)`
	`AddJob(78,1)`
	
	VB.NET
	`Sub Main()`
	`  Initialise()`
	`  AddJob(12, 10)`
	`  AddJob(526, 9)`
	`  AddJob(33, 8)`
	`  AddJob(12, 9)`
	`  AddJob(78, 1)`
	`End Sub`
		2

Question	Answer	Marks

1(e)	1 mark per point:
	
	• Procedure header (and end where appropriate)
	• Outer loop through all 5 elements / number of jobs …
	• … inner loop through array elements …
	• …and comparing **priority** (second index) …
	• …moving the elements up and inserting correctly
	
	Example program code:
	
	Python
	`def InsertionSort():`
	`  global Jobs`
	`  global NumberOfJobs`
	`  for I in range(1, NumberOfJobs):`
	`  Current1 = Jobs[I][0]`
	`  Current2 = Jobs[I][1]`
	`  while I > 0 and Jobs[I-1][1] > Current2:`
	`  Jobs[I][0] = Jobs[I-1][0]`
	`  Jobs[I][1] = Jobs[I-1][1]`
	`  I = I - 1`
	`  Jobs[I][0] = Current1`
	`  Jobs[I][1] = Current2`
	
	Java
	`public static void InsertionSort(){`
	`  Integer Current1;`
	`  Integer Current2;`
	`  Integer Counter;`
	`  Integer Placed;`
	`  for(Integer i = 1; i < NumberOfJobs; i++){`
	`  Current1 = Jobs[i][0];`
	`  Current2 = Jobs[i][1];`
	`  while(i > 0 && Jobs[i-1][1] > Current2){`
	`  Jobs[i][0] = Jobs[i-1][0];`
	`  Jobs[i][1] = Jobs[i-1][1];`
	`  i = i - 1;`
	
	`  }`
	`  Jobs[i][0] = Current1;`
	`  Jobs[i][1] = Current2;`
	`  }`
	`}`
		5

Question	Answer	Marks

1(e)	**VB.NET**
	`Sub InsertionSort()`
	`  Dim Tempa As Integer`
	`  Dim Tempb As Integer`
	`  Dim Counter As Integer`
	`  Dim Placed As Boolean`
	`  For i = 1 To NumberOfJobs - 1`
	`  Tempa = Jobs(i, 0)`
	`  Tempb = Jobs(i, 1)`
	`  Counter = i`
	`  Placed = False`
	`  While (Counter > 0 And Not Placed)`
	`  If (Jobs(Counter - 1, 1) > Tempb) Then`
	`  Jobs(Counter, 0) = Jobs(Counter - 1, 0)`
	`  Jobs(Counter, 1) = Jobs(Counter - 1, 1)`
	`  Counter = Counter - 1`
	`  Else`
	`  Placed = True`
	`  End If`
	`  End While`
	`  Jobs(Counter, 0) = Tempa`
	`  Jobs(Counter, 1) = Tempb`
	`  Next i`
	`End Sub`	

Question	Answer	Marks

1(f)	1 mark per point:
	
	• procedure heading (and end where appropriate) **and** outputting all job numbers and priorities
	• Outputting the job and priority for each element on the same line, with a line break between each job …
	• … with 'priority' between job number and priority
	
	Example program code:
	
	Java
	`public static void PrintArray(){`
	`  for(Integer x = 0; x < NumberOfJobs; x++){`
	`  System.out.println(Jobs[x][0] + " priority " + Jobs[x][1]);`
	`  }`
	`}`
	
	Python
	`def PrintArray():`
	`  global Jobs`
	`  global NumberOfJobs`
	`  for X in range(0, NumberOfJobs):`
	`  print(str(Jobs[X][0]), " priority ", str(Jobs[X][1]))`
	
	VB.NET
	`Sub PrintArray()`
	`  For X = 0 To NumberOfJobs - 1`
	`  Console.WriteLine(Jobs(X, 0) & " priority " & Jobs(X, 1))`
	`  Next`
	`End Sub`
		3

1(g)(i)	• calling both subroutines in the main program in the correct order
	
	Example program code:
	
	Java
	`InsertionSort();`
	`PrintArray();`
	
	Python
	`InsertionSort()`
	`PrintArray()`
	
	VB.NET
	`InsertionSort()`
	`PrintArray()`
		1

Question	Answer	Marks
1(g)(ii)	1 mark for added 5 times and jobs in order. 526 and 12 can be reversed Added Added Added Added Added 78 priority 1 33 priority 8 526 priority 9 12 priority 9 12 priority 10	**1**

Question	Answer	Marks

2(a)	1 mark per point:
	
	• Class declaration (and end where appropriate) for `Character`
	• Declaring the 3 private attributes with appropriate data types; `Name` as string, `xCoordinate` as integer, `yCoordinate` as integer
	• Constructor method (and end where appropriate) taking 3 parameters …
	• …assigning parameters to all 3 attributes
	
	Example program code:
	
	Java
	`class Character{`
	`  private String Name;`
	`  private Integer XCoordinate;`
	`  private Integer YCoordinate;`
	`  public Character(String Namep, Integer XCoord, Integer YCoord){`
	`  Name = Namep;`
	`  XCoordinate = XCoord;`
	`  YCoordinate = YCoord; }}`
	
	Python
	`class Character:`
	`  #private Name as string`
	`  #private XCoordinate as integer`
	`  #private YCoordinate as integer`
	`  def __init__(self, Namep, Xcoord, Ycoord):`
	`  self.__Name = Namep`
	`  self.__XCoordiante = Xcoord`
	`  self.__YCoordinate = Ycoord`
	
	VB.NET
	`Class Character`
	`  Private Name As String`
	`  Private XCoordinate As Integer`
	`  Private YCoordinate As Integer`
	`  Sub New(Namep, Xcoord, Ycoord)`
	`  Name = Namep`
	`  XCoordinate = Xcoord`
	`  YCoordinate = Ycoord`
	`  End Sub`
	`End Class`
		4

Question	Answer	Marks

2(b)	1 mark per point:
	
	• 1 get method header (and end where appropriate) with no parameters…
	• …returning correct value
	• 2nd and 3rd correct get methods
	
	Example program code:
	
	Java
	`public String GetName(){`
	`  return Name;}`
	`public Integer GetX(){`
	`  return XCoordinate;}`
	`public Integer GetY(){`
	`  return YCoordinate;}`
	
	Python
	`  def GetName(self):`
	`  return self.__Name`
	`  def GetX(self):`
	`  return self.__XCoordinate`
	`  def GetY(self):`
	`  return self.__YCoordinate`
	
	VB.NET
	`Function GetName()`
	`  Return Name`
	`End Function`
	`Function GetX()`
	`  Return XCoordinate`
	`End Function`
	`Function GetY()`
	`  Return YCoordinate`
	`End Function`
		3

Question	Answer	Marks

2(c)	1 mark per point:
	
	• method header (and end where appropriate) taking 2 (integer) parameters
	• adding both **parameters** to existing x and y coordinate values
	
	Example program code:
	
	Java
	`public void ChangePosition(Integer XChange, Integer YChange){`
	`  XCoordinate = XCoordinate + XChange;`
	`  YCoordinate = YCoordinate + YChange;`
	`}`
	Python
	`def ChangePosition(self, XChange, YChange):`
	`  self.__XCoordinate = self.__XCoordinate + XChange`
	`  self.__YCoordinate = self.__YCoordinate + YChange`
	
	VB.NET
	`Sub changePosition(XChange, YChange)`
	`  XCoordinate = XCoordinate + XChange`
	`  YCoordinate = YCoordinate + YChange`
	`End Sub`
		2

Question	Answer	Marks

2(d)	1 mark per point (**Max 7**):
	• declaration of 1D array, 10 elements of type `Character`
	• opening text file `Characters.txt` to read
	• looping until EOF/10 times…
	• … reading in **each** 3-set of values from file …
	• … instantiate a `Character` with correct parameters read in from file…
	• … store in next element/append in declared array
	• closing the text file (in appropriate place)
	• Exception handling for opening and reading data from file…
	• … with appropriate catch and output
	
	Example program code:
	
	Java
	`public static void main(String[] args){`
	`  Character[] Characters  = new Character[10];`
	`  String TextFile = "Characters.txt";`
	`  String Name = "";`
	`  Integer Xcoord = 0;`
	`  Integer Ycoord = 0;`
	`  try{`
	`  FileReader f = new FileReader(TextFile);`
	`  BufferedReader Reader = new BufferedReader(f);`
	`  for(Integer X = 0; X < 10; X++){`
	`  Name = Reader.readLine();`
	`  Xcoord = Integer.parseInt(Reader.readLine());`
	`  Ycoord = Integer.parseInt(Reader.readLine());`
	`  }`
	`  Reader.close();`
	`  }catch(FileNotFoundException ex){`
	`  System.out.println("No file found");`
	`  }`
	`  catch(IOException ex){`
	`  System.out.println("No file found");`
	`  }`
	`}`
	
	Python
	`Characters = []`
	`TextFile = "Characters.txt"`
	`try:`
	`  File = open(TextFile, 'r')`
	`  for X in range(0, 10):`
	`  Name = File.readline().strip()`
	`  XCoord = File.readline().strip()`
	`  YCoord = File.readline().strip()`
	`  TempC = Character(Name, int(XCoord), int(YCoord))`
	`  Characters.append(TempC)`
	`  File.close()`
	`except:`
	`  print("File not found")`
		7

Question	Answer	Marks

2(d)	**VB.NET**
	`Sub Main()`
	`  Dim Characters(0 To 9) As Character`
	`  Dim TextFile As String = "Characters.txt"`
	`  Try`
	`  Dim FileReader As New System.IO.StreamReader(TextFile)`
	`  For X = 0 To 10`
	`  Name = FileReader.ReadLine()`
	`  Xcoord = FileReader.ReadLine()`
	`  Ycoord = FileReader.ReadLine()`
	`  Characters(X) = New Character(Name, Xcoord, Ycoord)`
	`  Next`
	`  FileReader.Close()`
	`  Catch ex As Exception`
	`  Console.WriteLine("File not found")`
	`  End Try`
	`end sub`	

Question	Answer	Marks

2(e)	1 mark per point (**Max 5**):
	
	• Taking name as input …
	• …converting/checking case e.g. all to lower
	
	• Looping through array of characters comparing each character name to input …
	• …continuously taking repeat input if not found in array
	• …storing the index when found
	
	• Accessing the name of character in the array using `GetName()`
	
	Example program code:
	
	Python
	`Position = -1`
	`CharacterName = ""`
	`while(Position == -1):`
	`  CharacterInput = input("Enter the Character to move").rstrip('\n').lower()`
	`  for Count in range(0, 10):`
	`  Temp = str(Characters[Count].GetName().strip())`
	`  if(Temp == CharacterInput):`
	`  Position = Count`
	
	VB.NET
	`Dim Position As Integer = -1`
	`Dim CharacterName As String = ""`
	`While Position = -1`
	`  Console.WriteLine("Enter the Character to move")`
	`  CharacterName = Console.ReadLine`
	`  For Count = 0 To 9`
	`  If(Characters(Count).GetName).tolower = CharacterName.ToLower Then`
	`  Position = Count`
	`  End If`
	`  Next`
	`End While`
		5

Question	Answer	Marks

2(e)	**Java**
	`Integer Position = -1;`
	`String CharacterName = "";`
	`Scanner scanner = new Scanner(System.in);`
	`String Temp = "";`
	
	`while(Position == -1){`
	`  System.out.println("Enter the Character to move");`
	`  CharacterName = scanner.nextLine();`
	`  for(Integer Count = 0; Count < 10; Count++){`
	`  Temp = Characters[Count].GetName();`
	`  Temp = Temp.toLowerCase();`
	
	`  if(Temp.equals(CharacterName.toLowerCase())){`
	`  Position = Count;`
	`  } }}`	

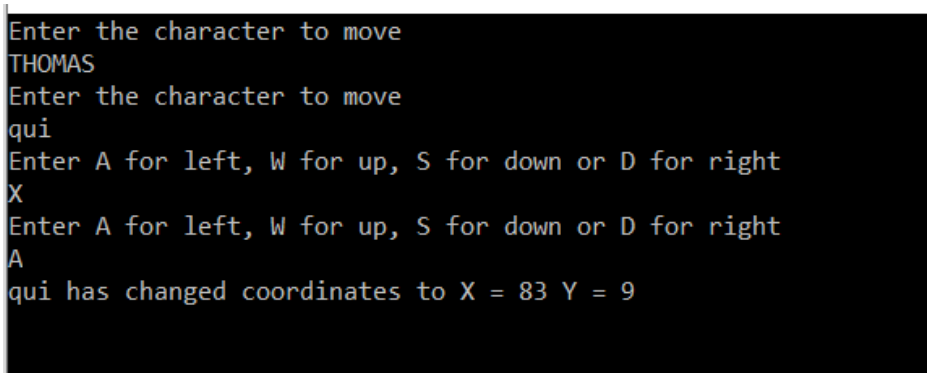
Question	Answer	Marks

2(f)	1 mark per point (**Max 7**):
	
	• Taking move as input…
	• …looping until valid
	• Calling `ChangePosition()` with object
	• If A is input **parameters** are −1, 0
	• If D is input **parameters** are 1, 0
	• If W is input **parameters** are 0, 1
	• If S is input **parameters** are 0, −1
	
	Example program code:
	
	Java
	`Boolean IsValid = false;`
	`String Move = "";`
	`while(IsValid != true){`
	`  System.out.println("Enter A for left, W for up, S or down or D for right");`
	`  Move = scanner.nextLine();`
	`  if(Move.toUpperCase().equals("A")){`
	`  Characters[Position].ChangePosition(-1,0);`
	`  IsValid = true;`
	`  } else if(Move.toUpperCase().equals("W")){`
	`  Characters[Position].ChangePosition(0,1);`
	`  IsValid = true;`
	`  } else if(Move.toUpperCase().equals("S")){`
	`  Characters[Position].ChangePosition(0,-1);`
	`  IsValid = true;`
	`  } else if(Move.toUpperCase().equals("D")){`
	`  Characters[Position].ChangePosition(1,0);`
	`  IsValid = true;`
	`  }  }`
	
	Python
	`IsValid = False`
	`while(IsValid != True):`
	`  Move = input("Enter A for left, W for up, S for down, or D for right")`
	`  if(Move.upper() == "A"):`
	`  Characters[Position].ChangePosition(-1,0)`
	`  IsValid = True`
	`  elif (Move.upper() == "W"):`
	`  Characters[Position].ChangePosition(0,1)`
	`  IsValid = True`
	`  elif (Move.upper() == "S"):`
	`  Characters[Position].ChangePosition(0,-1)`
	`  IsValid = True`
	`  elif(Move.upper() == "D"):`
	`  Characters[Position].ChangePosition(1,0)`
	`  IsValid = True`
		7

Question	Answer	Marks

2(f)	**VB.NET**
	`Dim IsValid As Boolean = False`
	`Dim Move As String`
	`While IsValid <> True`
	`  Console.WriteLine("Enter A for left, W for up, S for down or D for right")`
	`  Move = Console.ReadLine()`
	`  If Move.ToUpper = "A" Then`
	`  Characters(Position).ChangePosition(-1, 0)`
	`  IsValid = True`
	`  ElseIf Move.ToUpper = "W" Then`
	`  Characters(Position).ChangePosition(0, 1)`
	`  IsValid = True`
	`  ElseIf Move.ToUpper = "S" Then`
	`  Characters(Position).ChangePosition(0, -1)`
	`  IsValid = True`
	`  ElseIf Move.ToUpper = "D" Then`
	`  Characters(Position).ChangePosition(1, 0)`
	`  IsValid = True`
	`  End If`
	`End While`	

2(g)(i)	1 mark per point:
	• Outputting given message including name, x and y position
	• …all using appropriate get methods
	Example program code:
	Java
	`System.out.println(CharacterName + " has changed coordinates to X = " + Characters[Position].GetX() + " Y = " + Characters[Position].GetY());`
	Python
	`print(CharacterName, " has changed coordinate to X = ", str(Characters[Position].GetX()), " Y = ", str(Characters[Position].GetY()))`
	VB.NET
	`Console.WriteLine(CharacterName & " has changed coordinates to X = " & Characters(Position).GetX & " Y = " & Characters(Position).GetY())`
		2

Question	Answer	Marks
2(g)(ii)	1 mark for correct result, for example:  	**1**

Question	Answer	Marks

3(a)	1 mark per point:
	
	• 1D array with 100 (Integer) spaces
	• head pointer declared initialised to appropriate value e.g. −1
	• tail pointer declared initialised to 0
	
	Example program code:
	
	Java
	`public Integer[] queue = new Integer[100];`
	`public Integer HeadPointer = -1;`
	`public Integer TailPointer = 0;`
	
	Python
	`Queue = [-1 for I in range(100)] #Integer`
	`HeadPointer = -1`
	`TailPointer = 0`
	
	VB.NET
	`Dim Queue(0 To 99) As Integer`
	`Dim HeadPointer As Integer = -1`
	`Dim TailPointer As Integer = 0`
		3

Question	Answer	Marks

3(b)	1 mark per point:
	
	• Function header (and close where appropriate) with integer parameter
	• Checking if queue full **and** returning false
	• If not full adding **parameter** to queue at **tail pointer** …
	• … incrementing tail pointer (after adding to queue)
	• … and returning true
	• Changing head pointer to **0** if this is the first element in array
	
	Example program code:
	
	Java
	
	`public Boolean Enqueue(Integer Data){`
	`  if(TailPointer < 100){`
	`  if(HeadPointer == -1){`
	`  HeadPointer = 0;`
	`  }`
	`  Queue[TailPointer] = Data;`
	`  TailPointer = TailPointer + 1;`
	`  return true;`
	`  }`
	`  return false;`
	`}`
	Python
	`def Enqueue(Data):`
	`  global Queue`
	`  global TailPointer`
	`  if(TailPointer < 100):`
	`  if HeadPointer == -1:`
	`  HeadPointer = 0`
	`  Queue[TailPointer] = Data`
	`  TailPointer = TailPointer + 1`
	`  return True`
	`  return False`
	
	VB.NET
	`Function Enqueue(Data)`
	`  If TailPointer < 100 Then`
	`  If HeadPointer = -1 Then`
	`  HeadPointer = 0`
	`  End If`
	`  Queue(TailPointer) = data`
	`  TailPointer = TailPointer + 1`
	`  Return True`
	`  End If`
	`  Return False`
	`End Function`
		6

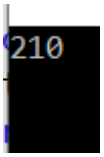
Question	Answer	Marks

3(c)	1 mark per point:
	
	• Looping 20 times
	• … using `Enqueue()` with each number 1 to 20 in ascending numerical order…
	• … and storing/using the return value
	• … based on return value, outputting "Successful" and "Unsuccessful" if **all** numbers are added
	
	Example program code:
	
	Java
	`public static void main(String[] args){`
	`  Boolean success = false;`
	`  for(Integer count = 1; count <= 20; count++){`
	`  success = enqueue(count);`
	`  }`
	`  if(success == false){`
	`  System.Out.Println("Unsuccessful ")`
	`  else{`
	`  System.Out.Println("Successful ")`
	`  }`
	`}`
	Python
	`Success = False`
	`for Count in range(1, 21):`
	`  Success = Enqueue(Count)`
	`if(Success == False):`
	`  print("Unsuccessful")`
	`else:`
	`  print("Successful")`
	
	VB.NET
	`Dim Success As Boolean`
	`For Count = 1 To 20`
	`  Success = Enqueue(Count)`
	`Next`
	`If Success = False THEN`
	`  Console.WriteLine("Unsuccessful")`
	`ELSE`
	`  Console.WriteLine("Successful")`
	`ENDIF`
		4

Question	Answer	Marks

3(d)	1 mark per point:
	
	• function call (and end where appropriate) taking a parameter
	• checking if at start of queue//20 …
	• …returning the last value in the queue
	• (otherwise) adding return value to a total // adding value in queue before recursive call **and** using this in the recursive call …
	• recursive call with Start/pointer −1
	• returning the final total
	
	Example program code:
	
	Java
	`public static Integer RecursiveOutput(Integer Start){`
	`  if(Start == 0){`
	`  return Queue[Start];`
	`  }else{`
	`  return Queue[Start] + RecursiveOutput(Start -1);`
	`  }}`
	
	Python
	`def RecursiveOutput(Start):`
	`  if(Start == 0):`
	`  return Queue[Start]`
	`  else:`
	`  return Queue[Start] + RecursiveOutput(Start - 1)`
	
	VB.NET
	`Function RecursiveOutput(ByVal Start)`
	`  If (Start = 0) Then`
	`  Return Queue(Start)`
	`  Else`
	`  Return Queue(Start) + RecursiveOutput(Start - 1)`
	`  End If`
	`End Function`
		6

3(e)(i)	1 mark for calling function **and** outputting return value.
	
	Example program code:
	
	Java
	`System.out.println(RecursiveOutput(TailPointer-1));`
	
	Python
	`print(str(RecursiveOutput(TailPointer - 1)))`
	
	VB.NET
	`Console.WriteLine(RecursiveOutput(TailPointer - 1))`
		1

Question	Answer	Marks
3(e)(ii)	1 mark for screenshot showing 210, for example: 	**1**