4.1 Abstraction and automation part 1

Name: ____________________

Class: ____________________

Date: ____________________

Time: **500 minutes**

Marks: **376 marks**

Comments:

Q1.

There are three boxes containing vegetables. One contains onions, one contains carrots and one contains onions and carrots. The three boxes have been labelled. One is labelled “onions”, one is labelled “carrots” and the other is labelled “onions and carrots”. You know that all three have been labelled incorrectly.

Describe how you can work out what each box actually contains by taking just **one** vegetable out of **one** box, without looking inside any of the boxes.

____

____

____

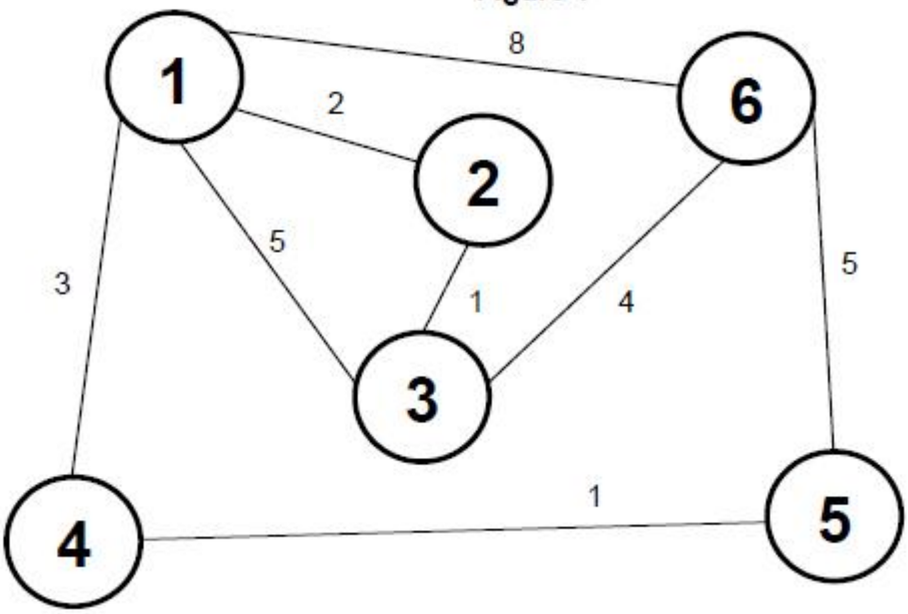
____

(Total 2 marks)

Q2.

Figure 1 is a graph that shows the time it takes to travel between six locations in a warehouse. The six locations have been labelled with the numbers 1 - 6. When there is no edge between two nodes in the graph this means that it is not possible to travel directly between those two locations. When there is an edge between two nodes in the graph the edge is labelled with the time (in minutes) it takes to travel between the two locations represented by the nodes.

Figure 1



(a) The graph is represented using an adjacency matrix, with the value 0 being used to indicate that there is no edge between two nodes in the graph.

A value should be written in every cell.

Complete the unshaded cells in **Table 1** so that it shows the adjacency matrix for **Figure 1**.

Table 1

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

(2)

(b) Instead of using an adjacency matrix, an adjacency list could be used to represent the graph. Explain the circumstances in which it would be more appropriate to use an adjacency list instead of an adjacency matrix.

____

____

____

____

(2)

(c) State **one** reason why the graph shown in **Figure 1** is **not** a tree.

____

____

(1)

(d) The graph in **Figure 1** is a weighted graph. Explain what is meant by a **weighted graph**.

____

____

(1)

Figure 2 contains pseudo-code for a version of Djikstra's algorithm used with the graph in **Figure 1**.

`Q` is a priority queue which stores nodes from the graph, maintained in an order based on the values in array `D`. The reordering of `Q` is performed automatically when a value in `D` is changed.

`AM` is the name given to the adjacency matrix for the graph represented in **Figure 1**.

Figure 2

```
Q ← empty queue
FOR C1 ← 1 TO 6
  D[C1] ← 20
  P[C1] ← -1
  ADD C1 TO Q
ENDFOR
D[1] ← 0
WHILE Q NOT EMPTY
  U ← get next node from Q
  remove U from Q
  FOR EACH V IN Q WHERE AM[U, V] > 0
    A ← D[U] + AM[U, V]
    IF A < D[V] THEN
      D[V] ← A
      P[V] ← U
    ENDIF
  ENDFOR
ENDWHILE
OUTPUT D[6]
```

(e) Complete the unshaded cells of **Table 2** to show the result of tracing the algorithm shown in **Figure 2**. Some of the trace, including the maintenance of Q, has already been completed for you.

Table 2

				D						P					
U	**Q**	**V**	**A**	**1**	**2**	**3**	**4**	**5**	**6**	**1**	**2**	**3**	**4**	**5**	**6**
-	1,2,3,4,5,6	-	-	20	20	20	20	20	20	-1	-1	-1	-1	-1	-1
				0											
1	2,3,4,5,6	2													
		3													
		4													
		6													
2	3,4,5,6	3													
3	4,5,6	6													
4	5,6	5													
5	6	6													
6	-														

(7)

(f) What does the output from the algorithm in **Figure 2** represent?

____

____ **(1)**

(g) The contents of the array `P` were changed by the algorithm. What is the purpose of the array `P`?

____

____

____

____ **(2)**

(Total 16 marks)

Q3.

Figure 1 shows the data Norbert, Phil, Judith, Mary, Caspar and Tahir entered into a binary search tree.

Figure 2 contains pseudo-code for a recursive binary tree search algorithm.

Figure 1

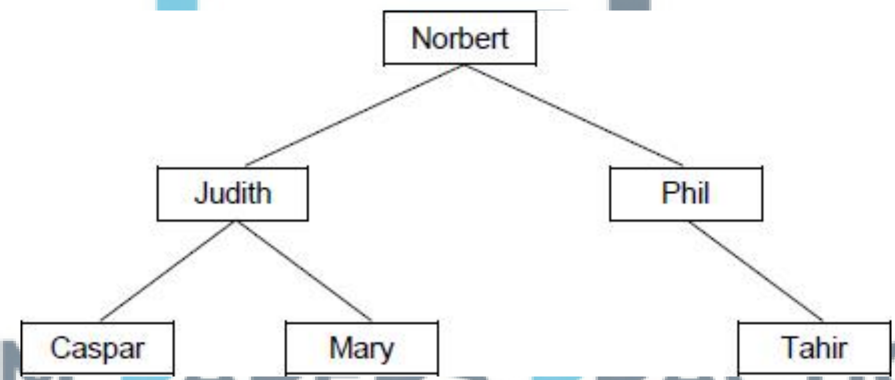


Figure 2

```
FUNCTION TreeSearch(target, node)
  OUTPUT 'Visited ', node
  IF target = node THEN
    RETURN True
  ELSE IF target > node AND Exists(node, right) THEN
    RETURN TreeSearch(target, node.right)
  ELSE IF target < node AND Exists(node, left) THEN
    RETURN TreeSearch(target, node.left)
  ENDIF
  RETURN False
ENDFUNCTION
```

The subroutine `Exists` takes two parameters – a node in the binary tree and a direction (`left` or `right`). It returns a Boolean value indicating if the node given as a parameter has a child node in the direction specified by the second parameter. For instance, `Exists(Mary, left)` will return a value of `False` as there is no node to the left of `Mary` in the binary tree.

`node.right` evaluates to the child node to the right of `node`, eg `Judith.right` is `Mary`.

`node.left` evaluates to the child node to the left of `node`, eg `Judith.left` is `Caspar`.

(a) What is meant by a recursive subroutine?

____

____

(1)

(b) There are two base cases for the subroutine `TreeSearch`. State **one** of the base cases.

____

____

(1)

(c) Complete the unshaded cells of the table below to show the result of tracing the `TreeSearch` algorithm shown in **Figure 2** with the function call `TreeSearch(Olivia, Norbert)`. You may not need to use all of the rows.

Function call	**Output**
`TreeSearch(Olivia, Norbert)`	

(3)

(Total 5 marks)

Q4.

State the name of an identifier for:

(a) an array or list variable

____

____

(1)

(b) a user-defined subroutine that has four parameters

____

____

(1)

(c) a variable that is used to store a whole number.

(1)

(d) a user-defined subroutine that returns one or more values.

(1)

(e) Look at the repetition structures in the `DisplayCavern` subroutine.

Explain the need for a nested `FOR` loop and the role of the `Count1` and `Count2` variables.

(3)

(f) Look at the `ResetCavern` subroutine.

Why has a named constant been used instead of the numeric value `5`?

(2)

(g) Look at the `SetPositionOfItem` subroutine.

Describe the purpose of the `WHILE` loop and the command within it in this subroutine.

(3)

(h) Look at the `MakeMonsterMove` subroutine.

Describe why it is necessary to check if the monster moves into the same cell as the flask and how any problem caused by this is solved by the **Skeleton Program**.

(3)

(i) Look at the `PlayGame` subroutine.

Explain why a `WHILE` loop has been made to complete the two moves for the monster rather than a `FOR` loop.

(2)

(j) The subroutines in the **Skeleton Program** avoid the use of global variables: they use local variables and parameter passing instead.

State **two** reasons why subroutines should, ideally, **not** use global variables.

(2)

(Total 19 marks)

Q5.

The famous detective John Stout was called in to solve a perplexing murder mystery. He determined the following facts.

a Nathan, the murdered man, was killed by a blow on the head.

b Either Suzanne or Martin was in the dining room at the time of the murder.

c If Peter was in the kitchen at the time of the murder, then Ian killed Nathan using poison.
d If Suzanne was in the dining room at the time of the murder, then Steve killed Nathan.
e If Peter was not in the kitchen at the time of the murder, then Martin was not in the dining room when the murder was committed.
f If Martin was in the dining room at the time the murder was committed, then Paul killed Nathan.
g If Kevin was in the hall at the time of the murder, then Suzanne killed Nathan by a blow to the neck with a saucepan.

(a) Who murdered Nathan?

A Paul
B Steve
C Suzanne
D Ian
E It is not possible for John Stout to solve the crime.

(1)

(b) Explain how you know your answer to (a) is correct.

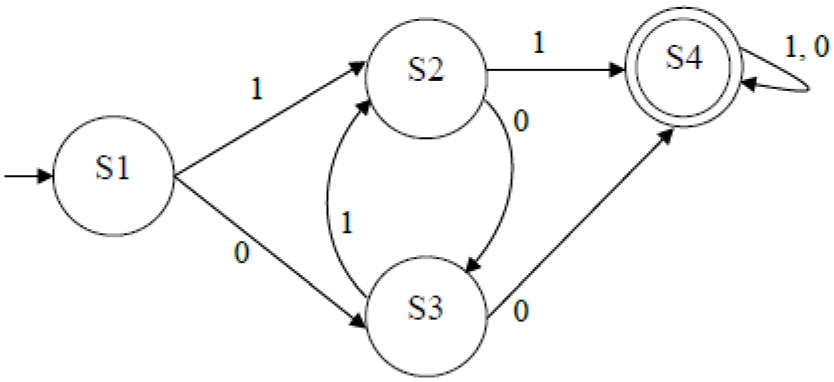
Use the space below for rough working.

(2)
(Total 3 marks)

Q6.

A finite state machine (FSM) can be used to define a language: a string is allowed in a language if it is accepted by the FSM that represents the rules of the language.
Figure 1 shows the state transition diagram for an FSM.

Figure 1



An FSM can be represented as a state transition diagram or as a state transition table. The table below is an incomplete state transition table for **Figure 1** .

(a) Complete the table.

Original state	**Input**	**New state**
S3		
S3		

(1)

(b) Any language that can be defined using an FSM can also be defined using a regular expression.

The FSM in **Figure 1** defines the language that allows all strings containing at least, either two consecutive 1s or two consecutive 0s.

The strings 0110, 00 and 01011 are all accepted by the FSM and so are valid strings in the language.

The strings 1010 and 01 are not accepted by the FSM and so are not valid strings in the language.

Write a regular expression that is equivalent to the FSM shown in **Figure 1**.

____

____

____

____

____

____

(3)

(c) Backus-Naur Form (BNF) can be used to define the rules of a language.

Figure 2 shows an attempt to write a set of BNF production rules to define a language of full names.

Figure 2

Note: underscores (_) have been used to denote spaces.
Note: rule numbers have been included but are not part of the BNF rules.

Rule number

```
1    <fullname> ::= <title>_<name>_<endtitle> |
                    <name> |
                    <title>_<name> |
                    <name>_<endtitle>

2    <title> ::= MRS | MS | MISS | MR | DR | SIR

3    <endtitle> ::= ESQUIRE | OBE | CBE

4    <name> ::= <word> |
                <name>_<word>

5    <word> ::= <char><word>

6    <char> ::= A | B | C | D | E | F | G | H | I |
                J | K | L | M | N | O | P | Q | R |
                S | T | U | V | W | X | Y | Z
```

BNF can be used to define languages that are not possible to define using regular expressions. The language defined in **Figure 2** could not have been defined using regular expressions.

Complete the table below by writing either a **'Y'** for **Yes** or **'N'** for **No** in each row.

Rule number (given in Figure 2)	**Could be defined using a regular expression**
1	
2	
3	
4	
5	
6	

(1)

(d) There is an error in rule 5 in **Figure 2** which means that no names are defined by the language.

Explain what is wrong with the production rule and rewrite the production rule so that the language does define some names – the names 'BEN D JONES', 'JO GOLOMBEK' and 'ALULIM' should all be defined.

____

____

____

____

(2)
(Total 7 marks)

Q7.

A computer program is being developed to play a card game on a smartphone. The game uses a standard deck of 52 playing cards, placed in a pile on top of each other.

The cards will be dealt (ie given out) to players from the top of the deck.

When a player gives up a card it is returned to the bottom of the deck.

(a) Explain why a queue is a suitable data structure to represent the deck of cards in this game.

____

____

(1)

(b) The queue representing the deck of cards will be implemented as a **circular** queue in a fixed size array named `DeckQueue`. The array `DeckQueue` has indices running from `1` to `52`.

The figure below shows the contents of the `DeckQueue` array and its associated pointers at the start of a game. The variable `QueueSize` indicates how many cards are currently represented in the queue.

DeckQueue

Index	Data
[1]	10-Hearts
[2]	2-Spades
[3]	King-Hearts
[4]	Ace-Clubs
. . .	
[52]	8-Diamonds

`FrontPointer = 1`

`RearPointer = 52`

`QueueSize = 52`

(i) Ten cards are dealt from the top of the deck.

What values are now stored in the `FrontPointer` and `RearPointer` pointers and the `QueueSize` variable?

`FrontPointer =` ___________ `RearPointer =` ___________

`QueueSize =` ___________

(1)

(ii) Next, a player gives up two cards and these are returned to the deck.

What values are now stored in the `FrontPointer` and `RearPointer` pointers and the `QueueSize` variable?

`FrontPointer =` ___________ `RearPointer =` ___________

`QueueSize =` ___________

(1)

(iii) Write a pseudo-code algorithm to deal a card from the deck.

Your algorithm should output the value of the card that is to be dealt and make any required modifications to the pointers and to the `QueueSize` variable.

It should also cope appropriately with any situation that might arise in the `DeckQueue` array whilst a game is being played.

____

____

____

____

____

____

____

____

____

(6)

(c) The program for the card game will be an event-driven program.

Explain what it means for a program to be described as event-driven.

____

____

____

____

____

(2)

(d) The card game program will interact with the operating system on the smartphone.

Describe **two** differences between the operating system that is installed on the smartphone and an operating system that would be used on a desktop computer.

Difference 1 ____

____

____

Difference 2 ____

____

____

(2)

(Total 13 marks)

Q8.

A computer program is being developed to allow commuters to plan journeys on the London Underground railway network which connects together over 250 stations.

The program needs to store a representation of the network so that the **shortest route** (ie shortest distance) between any two stations can be found.

Figure 1 is a map of central London, showing the location of ten of the stations on the London Underground. The locations of the underground railway lines are not shown. Note that nine of the stations are indicated by the symbol ⊖ but Charing Cross has a different symbol ≠ because it is a combined underground and overground station.

Figure 1

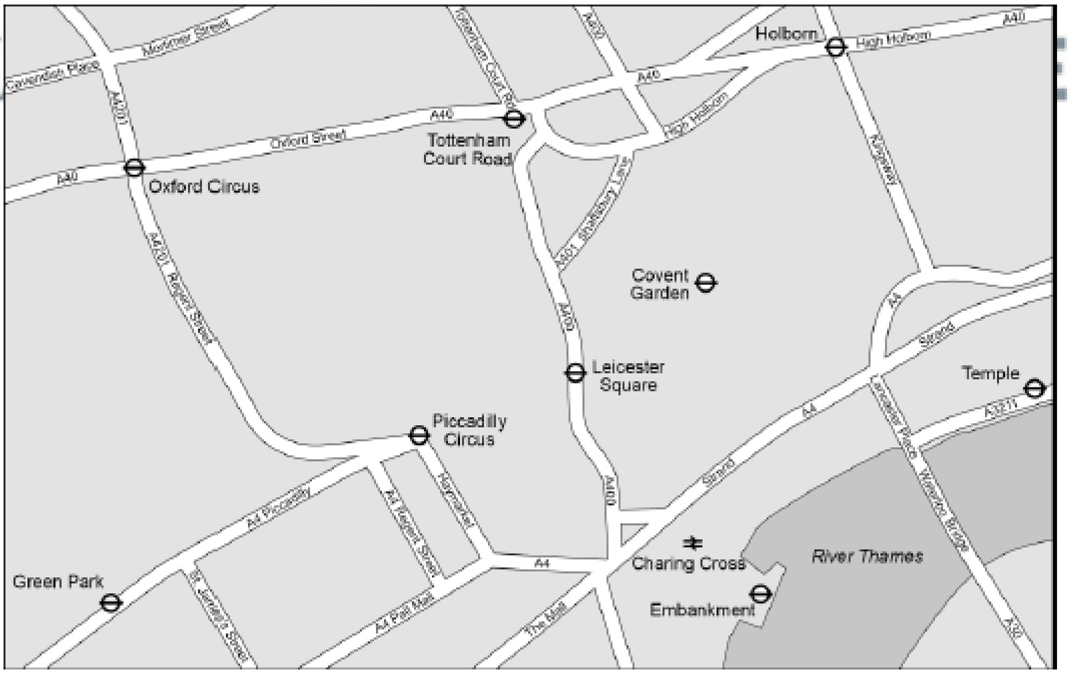


Figure 2 is a map of part of the underground railway network, showing the same ten stations. This map does not show the streets above ground but instead shows the

underground railway lines that connect the stations together.

Figure 2

Due to copyright restrictions we are unable to show this image. Please use the link below to find the appropriate section of the tube map.

Standard Tube map - Transport for London

Figure 2 can be used in conjunction with a table of distances between adjacent stations to calculate the shortest route between any two stations on the network.

The map of the entire underground railway network (**not** just the parts shown in **Figure 1** and **Figure 2**) together with the full table of distances can be represented logically as a graph.

(a) The representation of the underground railway network as a graph is an abstraction.

Explain what an abstraction is.

____

____

____

(1)

(b) Write a detailed description of:

- how the underground railway network and table of distances could be represented as a graph, **and**,
- how this representation could be implemented as either an adjacency matrix **or** an adjacency list (describe **one** of these alternatives only), using array(s) in a programming language that does not have a built-in data structure for graphs.

Your implementation should store all the details that are required to calculate the shortest distance between any two stations, but you do not need to describe how the shortest distance would be worked out.

In your answer you will be assessed on your ability to use good English, and to organise your answer clearly in complete sentences, using specialist vocabulary where appropriate.

You may use diagrams to help clarify your description, but as you are being assessed on your ability to use good English, you must ensure that all diagrams are fully explained.

____

____

____

(8)
(Total 9 marks)

Q9.

An interactive operating system maintains a list of the processes that are currently waiting to execute (run). The processes are stored in order of the priority that is associated with their execution. This priority can be set as "High", "Normal" or "Low".

Figure 1 and **Figure 2** below show two different ways in which the storage of the process list could be implemented.

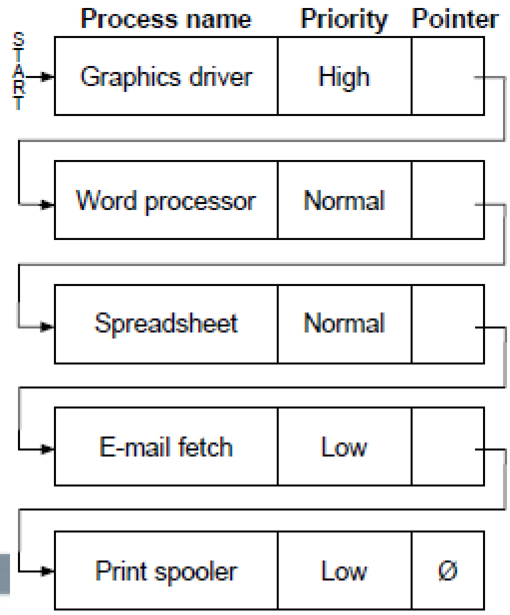
Figure 1

Static implementation: as an ordered list using a fixed size array.

Index	Process name	Priority
[1]	Graphics driver	High
[2]	Word processor	Normal
[3]	Spreadsheet	Normal
[4]	E-mail fetch	Low
[5]	Print spooler	Low
⋮		
[100]		

Figure 2

Dynamic implementation: as a linked list using dynamic memory allocation.



The process at the start of the list will be run next. In **Figure 1** and **Figure 2**, this is the "Graphics driver" process.

When a new process is initiated it is inserted into the list immediately after the last process of the same priority. A "Computer game" process with "High" priority would be inserted into the list in **Figure 1** and **Figure 2** between the "Graphics driver" and "Word processor" processes.

When a process is completed it is deleted from the list.

(a) Explain **two** differences between a dynamic data structure and a static data structure.

Difference 1: ____

____

Difference 2: ____

____

(2)

(b) The **static implementation** is less efficient at inserting new items into the list than the **dynamic implementation**.

Explain why this is the case.

____

____

(2)

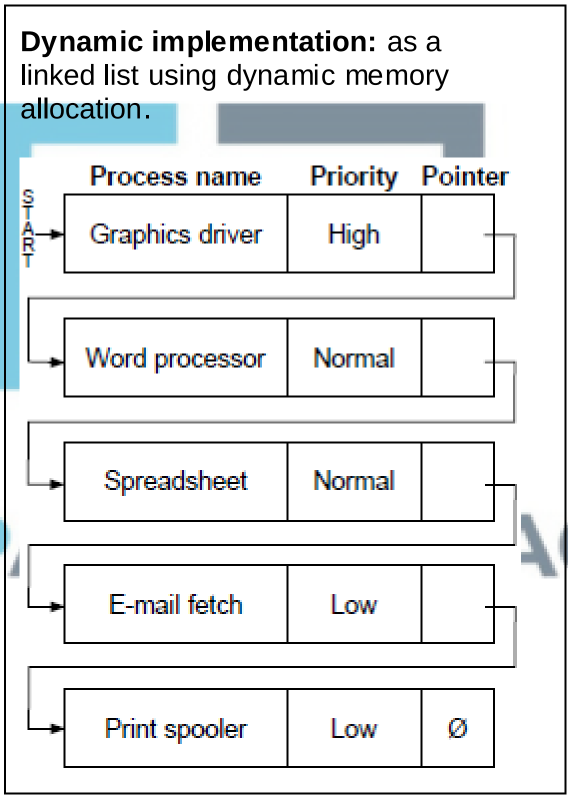
(c) At a higher level of abstraction, the process list maintained by the operating system could be viewed as a type of queue.

What type of queue?

(1)

Figure 2 is repeated below so that you can answer the remaining question parts without having to turn back in the question booklet.

Figure 2 (repeated)



(d) Consider the **dynamic implementation** in **Figure 2**.

(i) What will the heap be used for in this implementation?

(1)

(ii) In **Figure 2** pointers are shown as arrows.

When the linked list is created in a programming language, what will the integer value stored in a pointer represent?

(1)

(iii) Write an algorithm of the steps that would be involved in inserting a new process "Database" with priority "Normal" into the dynamic implementation linked list in **Figure 2**.

The algorithm will need to:

- find the correct position to insert the new process at, **then**
- make the necessary changes to insert the information about the new process.

You may wish to use a **Current Node Pointer** and a **Previous Node Pointer** in your response.

Your algorithm only needs to cater for a list that already contains some processes at each priority level.

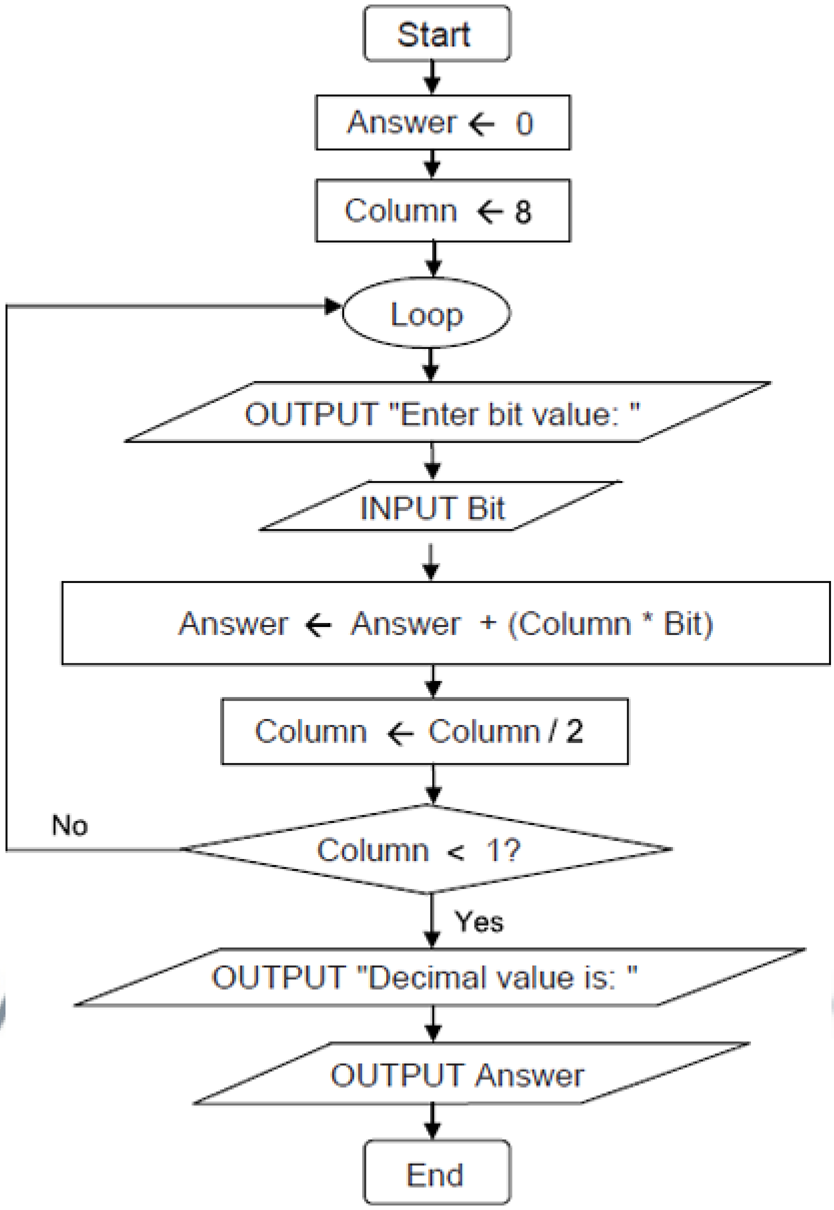
(7)

(Total 14 marks)

Q10.

Create a folder/directory for your new program.

The algorithm, represented as a flowchart below, and the variable table, describe the converting of a 4-bit binary value into denary.



Identifier	**Data type**	**Purpose**
`Column`	`Integer`	Stores the place value (column heading)
`Answer`	`Integer`	Stores the denary value equivalent to the bit pattern entered by the user
`Bit`	`Integer`	Stores a `0` or `1` entered by the user

What you need to do

Write a program for the above algorithm.

Test the program by showing the result of entering the values `1, 1, 0, 1 (in that order).`

Save the program in your new folder/directory.

Evidence that you need to provide

(a) Your PROGRAM SOURCE CODE.

(11)

(b) SCREEN CAPTURE(S) for the test described above.

(3)

(c) What is the largest denary number that could be output by the algorithm represented by the flowchart in the diagram above?

____

(1)

(d) The algorithm represented by the flowchart above can convert sixteen different bit patterns into denary.

If the symbol $\boxed{\text{Column} \leftarrow 8}$ is changed to $\boxed{\text{Column} \leftarrow 16}$ how many **more** bit patterns could be converted into denary?

____

(1)

(e) When developing a new system the stages of the systems development life cycle could be followed.

At which stage of the systems development life cycle would the flowchart above have been created?

____

(1)

(f) At which stage of the systems development life cycle would the algorithm represented by the flowchart above be automated using a programming language?

____

(1)

(Total 18 marks)

Q11.

State the name of an identifier for:

(a) a user-defined subroutine that has only one parameter.

____

(1)

(b) user-defined subroutine whose only action is to produce output to the screen.

(1)

(c) a variable that has a stepper role.

(1)

(d) an array variable.

(1)

(e) Look at the repetition structure in the `SetPositionOfItem` subroutine.

Describe the circumstances under which this structure in the **Skeleton Program** will stop repeating.

(3)

(f) Look at the `SetUpGame` subroutine.

Why has a `For` loop been chosen for the repetition structure?

(1)

(g) The `For` loop repeats `NoOfTrap` times.

Why has a named constant ben used instead of the numeric value 2?

(1)

(h) When a game is saved it is stored as a binary file. A text file could have been used instead.

Describe a difference between the way that data are stored in a binary file and the way that data are stored in a text file.

(2)

(i) The subroutines in the **Skeleton Program** avoid the use of global variables – they use local variables and parameter passing instead.

State **two** reasons why subroutines should, ideally, not use global variables.

(2)

(j) Below is a pseudo-code representation of the part of the `PlayGame` subroutine that is used to check if the player has triggered one of the traps in the cavern.

```
MonsterAwake ← CheckIfSameCell(PlayerPosition,
                  TrapPositions[1])
If Not MonsterAwake
   Then MonsterAwake ← CheckIfSameCell(PlayerPosition,
                  TrapPositions[2])
EndIf
```

Why is it necessary that the check for the triggering of the second trap is inside the selection structure?

(2)

(Total 15 marks)

Q12.

(a) Time complexity is one of the two measures that are used to describe the complexity of an algorithm.

What is the other measure?

(1)

(b) A student has been asked to write a program to list duplicate entries in a file containing a list of words. The diagram below shows her first attempt at planning an algorithm. The algorithm will not work in all circumstances.

```
Open file
N ← Number of items in file
```

```
For Pos1 ← 1 To N Do
  Read item at position Pos1 in file into variable W1
  For Pos2 ← 1 To N Do
    Read item at position Pos2 in file into variable W2
    If W1 = W2 And Not (Pos1 = Pos2)
      Then Output 'Duplicate: ' , W1
    EndIf
  EndFor
EndFor
Close file
```

The basic operation in the algorithm is the `If` statement that compares two words. The contents of a particular file are shown in the table below.

File position	**Item**
1	Rope
2	Dagger
3	Rope

(i) Complete the table below by tracing the execution of the algorithm in the diagram above when it is applied to the file in the table above.

N	**Pos1**	**W1**	**Pos2**	**W2**	**Output**

(3)

(ii) Tick **one** box in the table below to indicate the correct order of time complexity of the algorithm that the student has written.

Order of time complexity	Tick one box
$O(a^n)$	
$O(n)$	
$O(n^2)$	

(1)

(iii) Justify your answer to part (ii).

____

____

____

(2)
(Total 7 marks)

Q13.

A graph can be drawn to represent a maze. In such a graph, each graph vertex represents one of the following:

- the entrance to or exit from the maze
- a place where more than one path can be taken
- a dead end.

Edges connect the vertices according to the paths in the maze.

Diagram 1 shows a maze and **Diagram 2** shows one possible representation of this maze.

Position 1 in **Diagram 1** corresponds to vertex 1 in **Diagram 2** and is the entrance to the maze. Position 7 in **Diagram 1** is the exit to the maze and corresponds to vertex 7.

Dead ends have been represented by the symbol in **Diagram 2**.

Diagram 3 shows a simplified undirected graph of this maze with dead ends omitted.

Diagram 1

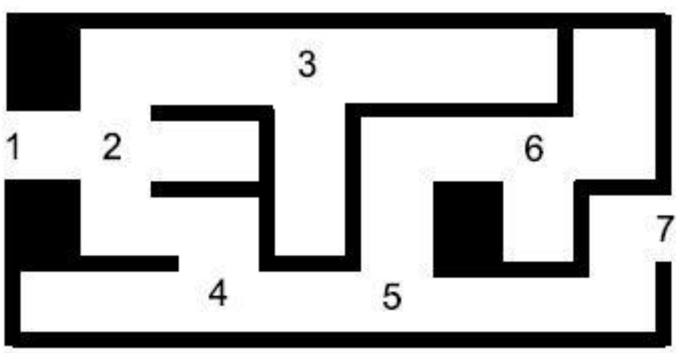
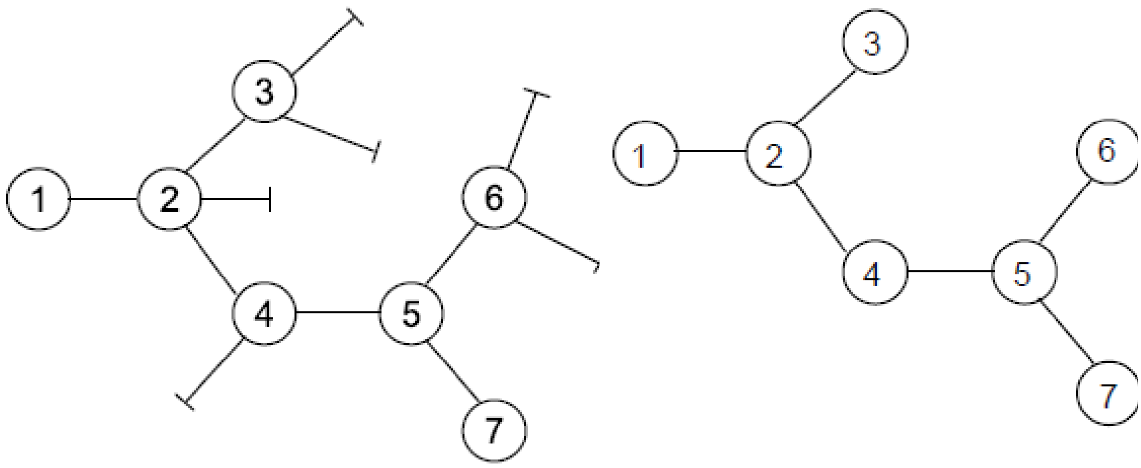


Diagram 2 **Diagram 3**



Representation of maze including dead ends

Graph representing maze with dead ends omitted

(a) The graph in **Diagram 3** is a tree.

State **one** property of the graph in **Diagram 3** that makes it a tree.

____

____

(1)

(b) The graphs of some mazes are not trees.

Describe a feature of a maze that would result in its graph **not** being a tree.

____

____

(1)

(c) Complete the table below to show how the graph in **Diagram 3** would be stored using an adjacency matrix.

(2)

(d) (i) What is a *recursive routine?*

____

____

(1)

(ii) To enable the use of recursion a programming language must provide a stack.

Explain what this stack will be used for and why a stack is appropriate.

____

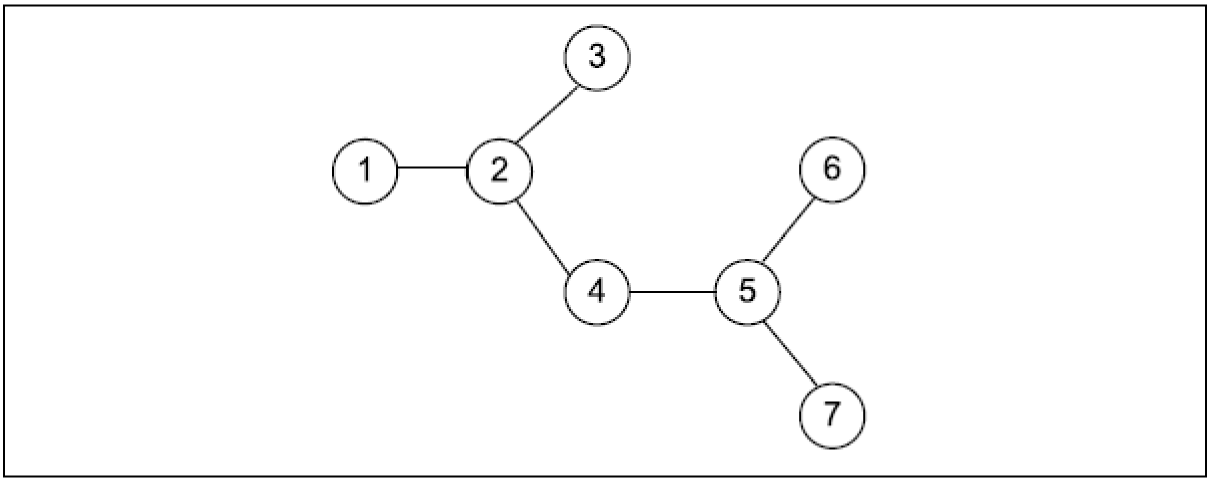
____

____

____

(2)

Diagram 3 is repeated here so that you can answer Question (e) without having to turn pages.



(e) A recursive routine can be used to perform a depth-first search of the graph that represents the maze to test if there is a route from the entrance (vertex 1) to the exit (vertex 7).

The recursive routine in the diagram below is to be used to explore the graph in **Diagram 3**. It has two parameters, `V` (the current vertex) and `EndV` (the exit vertex).

```
Procedure DFS(V, EndV)
    Discovered[V] ← True
    If V = EndV Then Found ← True
    For each vertex U which is connected to V Do
        If Discovered [U] = False Then DFS(U, EndV)
    EndFor
    CompletelyExplored[V] ← True
EndProcedure
```

Complete the trace table below to show how the `Discovered` and `CompletelyExplored` flag arrays and the variable `Found` are updated by the algorithm when it is called using `DFS(1,7)`.

The details of each call and the values of the variables `V`, `U` and `EndV` have already been entered into the table for you. The letter `F` has been used as an abbreviation for `False`. You should use `T` as an abbreviation for `True`.

				Discovered							CompletelyExplored							
Call	V	U	EndV	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	Found
	-	-		F	F	F	F	F	F	F	F	F	F	F	F	F	F	F
DFS(1,7)	1	2	7															
DFS(2,7)	2	1	7															
		3	7															
DFS(3,7)	3	2	7															
DFS(2,7)	2	4	7															
DFS(4,7)	4	2	7															
		5	7															
DFS(5,7)	5	4	7															
		6	7															
DFS(6,7)	6	5	7															
DFS(5,7)	5	7	7															
DFS(7,7)	7	5	7															
DFS(5,7)	5	-	7															
DFS(4,7)	4	-	7															
DFS(2,7)	2	-	7															
DFS(1,7)	1	-	7															

(5)

(Total 12 marks)

Q14.

Bob has a problem that he needs to solve. The problem is described below.

"*There are two jugs – A and B. Jug A has a capacity of three litres. Jug B has a capacity of five litres. There are no markings on the jugs, so it is not possible to tell exactly how much is in a jug just by looking (unless it is full or empty). There is a sink with a water tap and a drain. How can exactly one litre of water be obtained from the tap using the two jugs*?"

A well-defined problem consists of a given, a goal, a set of resources, a set of constraints and ownership.

(a) Describe the *goal* of this problem.

____

____

(1)

(b) Describe the set of *resources* available to Bob when solving this problem.

____

(3)

(c) What is meant by *ownership* of a problem?

(1)

(Total 5 marks)

Q15.

A constant is a value that does not change throughout a program. Instead of referring to the value itself throughout a program, a named constant can be used.

(a) Give an example of a constant declaration from the **Skeleton Program**.

(1)

(b) State **one** advantage of using named constants for constant values.

(1)

(c) State the name of an identifier for a variable that has a fixed value role.

(1)

(d) State the name of an identifier for a variable that has a most wanted holder role.

(1)

The decision table shown below represents the logic of the selection structure in the `GetMenuChoice` subroutine. '✓' has been used to indicate the action that results from particular values for the conditions. The decision table is only partially complete; some incomplete parts have been labelled **(a), (b), (c)** and **(d)**

Conditions	`OptionChosen <1`	`True`	`False`	`False`	`False`
	`OptionChosen > 4`	`False`	`True`	`True`	**(d)**

	`OptionChosen <> 9`	**(c)**	`False`	`True`	`True`
Action	Output error message	✓	**(a)**	**(b)**	

(e) Which of the two cells labelled **(a)** and **(b)** should have a '✓' in it?

____ **(1)**

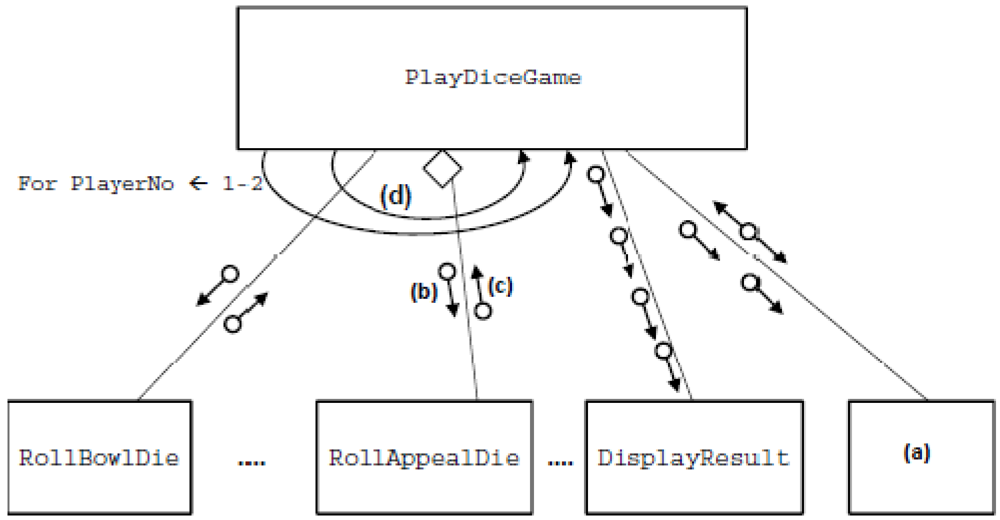
(f) What should be the contents of the cell labelled **(c)**?

____ **(1)**

(g) What should be the contents of the cell labelled **(d)**?

____ **(1)**

The diagram below shows an incomplete structure chart for part of the **Skeleton Program.**



With reference to the **Skeleton Program** and the diagram above, answer questions h to k.

(h) What should be written in box **(a)** in figure above?

____ **(1)**

(i) How should the arrow **(b)** in the diagram above be labelled?

____ **(1)**

(j) How should the arrow **(c)** in the diagram above be labelled?

____ **(1)**

(k) How should the curved arrow **(d)** in the diagram above be labelled?

____

(1)

(l) There is a variable called `Count` in the `LoadTopScores` subroutine. There is also a variable called `Count` in the `UpdateTopScores` subroutine.

Explain why these two different variables can have the same identifier.

____

____

____

____

(2)

(m) Look at the repetition structure in the `UpdateTopScores` subroutine, used to find the lowest of the current top scores.

When `UpdateTopScores` is called, how many times will this section of code repeat?

____

(1)

(n) Describe what the selection structure inside the repetition structure does.

____

____

____

____

____

____

(4)

(Total 18 marks)

Q16.

Reverse Polish Notation is an alternative to standard infix notation for writing arithmetic expressions.

(a) Convert the following Reverse Polish Notation expressions to their equivalent infix expressions.

Reverse Polish Notation	Equivalent Infix Expression
45 6 +	
12 19 + 8 *	

(2)

(b) State **one** advantage of Reverse Polish Notation over infix notation.

____

____

(1)

(c) The pseudo-code algorithm below can be used to calculate the result of evaluating a Reverse Polish Notation expression that is stored in a string. The algorithm is designed to work only with the single digit denary numbers 0 to 9. It uses procedures and functions listed in the table below, two of which operate on a stack data structure.

```
StringPos ← 0
Repeat
  StringPos ← StringPos + 1
  Token ← GetCharFromString(InputString, StringPos)
  If Token = '+' Or Token = '-' Or Token = '/' Or Token = '*'
    Then
      Op2 ←Pop()
      Op1 ← Pop()
      Case Token Of
        '+': Result ← Op1 + Op2
        '-': Result ← Op1 - Op2
        '/': Result ← Op1 / Op2
        '*': Result ←Op1 * Op2
      EndCase
      Push(Result)
    Else
      IntegerVal ←ConvertToInteger(Token)
      Push(IntegerVal)
  EndIf
Until StringPos = Length(InputString)
Output Result
```


Procedure/Function	Purpose	Example(s)
GetCharFromString (InputString:String , StringPos:Integer): Char	Returns the character at position `StringPos` within the string `InputString`. Note that the leftmost letter is position 1, not position 0.	`GetCharFromString ("Computing", 1)` would return the character `'C'`. `GetCharFromString ("Computing", 3)` would return the character `'m'`.
ConvertToInteger (ACharacter: Char): Integer	Returns the integer equivalent of the character in `ACharacter`.	`ConvertToInteger('4')` would return the integer value `4`.
Length (AString: String): Integer	Returns a count of the number of characters in the string `AString`.	`Length("AQA")` would return the integer value `3`.
Push (ANumber: Integer)	Puts the number in `ANumber` onto the stack.	`Push(6)` would put the number 6 on top of the

		stack.
Pop (): Integer	Removes the number from the top of the stack and returns it.	X ← Pop() would remove the value from the top of the stack and put it in X.

(d) Complete the table below to trace the execution of the algorithm when `InputString` is the string: `64+32+*`

In the `Stack` column, show the contents of the stack once for each iteration of the `Repeat..Until` loop, as it would be at the end of the iteration.

The first row and the leftmost column of the table have been completed for you.

StringPos	**Token**	**IntegerVal**	**Op1**	**Op2**	**Result**	**Stack**
0	-	-	-	-	-	
1						
2						
3						
4						
5						

6						
7						

(5)

Final output of algorithm: ____

(1)

(e) A programmer is going to implement the algorithm above in a programming language that does not provide built-in support for a stack data structure.

The programmer intends to simulate a stack by using a fixed length array of 20 integers named `StackArray` with indices running from `1` to `20` and an integer variable `TopOfStackPointer` which will be initialised to `0`.

Write a pseudo-code algorithm for the `Push` operation to push a value stored in the variable `ANumber` onto the stack.

Your algorithm should cope appropriately with any potential errors that might occur.

(4)

(Total 13 marks)

Q17.

(a) Explain what is meant by an *algorithm.*

____

____

(2)

(b) One way of checking that an algorithm is correct is to complete a dry run.

Dry run the algorithm in the figure below by completing the table below.

Assume that x has a value of 7.
The `MOD` operator calculates the remainder resulting from an integer division.

```
Answer ← True
FOR Count ← 2 To (x - 1) DO
    Remainder ← x MOD Count
    IF Remainer = 0 THEN
        Answer ← False
    ENDIF
ENDFOR
```

Answer	**Count**	**Remainder**
True	–	–
	2	1

(6)

(c) What is the purpose of this algorithm?

____

____

(1)
(Total 9 marks)

Q18.

A particular Turing machine has states S_1, S_2 and S_3.
S_1 is the start state and S_3 is the stop state.
The machine uses one tape which is infinitely long in one direction to store data.
The machine's alphabet is 0, 1, o, e and □, where □ is the symbol used to indicate a blank cell on the tape.

The transition rules for this Turing machine can be expressed as a transition function δ.
Rules are written in the form:

δ(Current State, Input Symbol) = (Next State, Output Symbol, Movement)

So, for example, the rule:

$$\delta\,(S_1, 0) = (S_1, 0, \rightarrow)$$

means

IF the machine is currently in state S_1 AND the input symbol read from the tape is 0
THEN the machine should remain in state S_1, write a 0 to the tape and move the read/write head one cell to the right

The machine's transition function, δ, is defined by:

$$\delta\,(S_1, 0) = (S_1, 0, \rightarrow)$$

$$\delta\,(S_1, 1) = (S_2, 1, \rightarrow)$$

$$\delta\,(S_1, \square) = (S_3, e, \rightarrow)$$

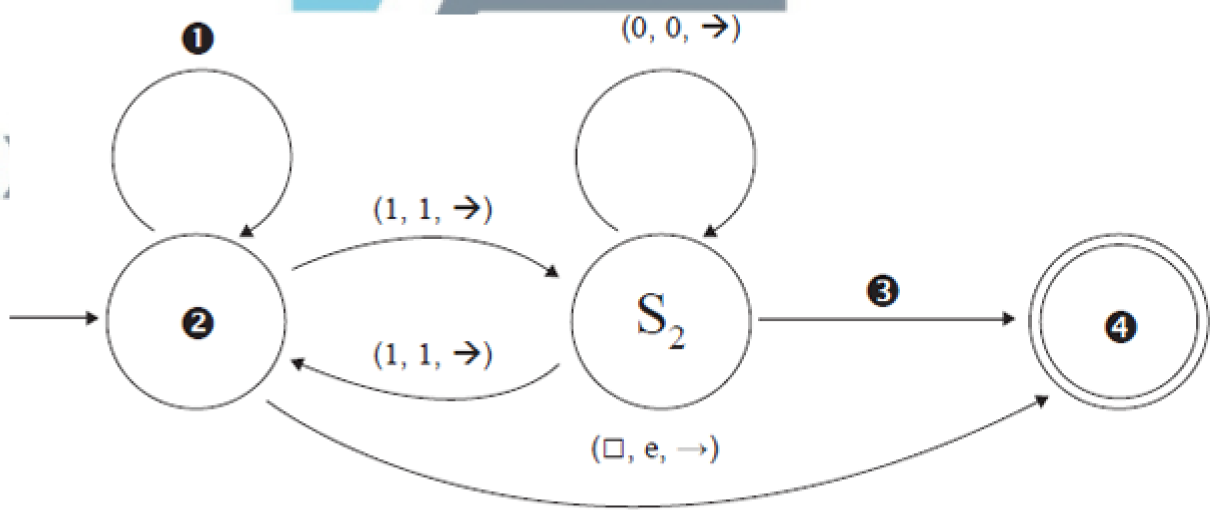
$$\delta\,(S_2, 0) = (S_2, 0, \rightarrow)$$

$$\delta\,(S_2, 1) = (S_1, 1, \rightarrow)$$

$$\delta\,(S_2, \square) = (S_3, 0, \rightarrow)$$

The diagram below shows a partially labelled finite state transition diagram for this machine.

Some labels are missing and have been replaced by numbers such as ❶. Each state transition arrow is labelled with the input symbol, the output symbol and the direction of movement, in that order. For example (□, e, →) means that if the input symbol is □, an e is written to the tape and the read/write head moves right one cell.



(a) Four labels are missing from the diagram above.

Write the missing labels in the table below.

Number	Correct Label
❶	

❷	
❸	
❹	

(2)

(b) The Turing machine is carrying out a computation using one tape which is infinitely long in one direction. The machine starts in state S_1 with the string `01100` on the tape.
All other cells contain the blank symbol, □. The read/write head is positioned at the leftmost zero, as indicated by the arrow.

			0	1	1	0	0		
			↑						

Current State: $\boxed{S_1}$

Trace the computation of the Turing machine, using the transition function δ. Show the contents of the tape, the current position of the read/write head and the current state as the input symbols are processed.

Current State: ☐

Current State: ☐

Current State: ☐

Current State: ☐

Current State: ☐

Current State: ☐

(4)

(c) What is the purpose of the algorithm represented by this Turing machine?

____

____

____

(1)

(d) Explain the importance of the theory of Turing machines to the subject of computation.

____

____

____

____

____

(2)
(Total 9 marks)

Q19.

A list data structure can be represented using an array.

The pseudocode algorithm in the diagram below can be used to carry out one useful operation on a list.

```
p ← 1
If ListLength > 0 Then
    While p <= ListLength And List [p]

        p ← p + 1
    EndWhile

    For q ← ListLength DownTo p Do
        List[q + 1] := List[q]
    EndFor
EndIf

List[p] ← New

ListLength ← ListLength + 1
```


(a) The initial values of the variables for one particular execution of the algorithm are shown in the trace table below.

Complete the trace table for the execution of the algorithm.

				List				
ListLength	New	p	q	[1]	[2]	[3]	[4]	[5]
4	38	-	-	9	21	49	107	

(4)

(b) Describe the purpose of the algorithm the diagram above.

(1)

(c) A list implemented using an array is a static data structure. The list could be implemented using a linked list as a dynamic data structure instead.

(i) Describe **one** difference between a static data structure and a dynamic data structure.

(1)

(ii) If the list were to be implemented as a dynamic data structure, explain what the heap would be used for.

(1)

(Total 7 marks)

Q20.

(a) (i) Explain what is meant by a pixel.

(1)

(ii) How are pixels encoded to form a bitmapped image?

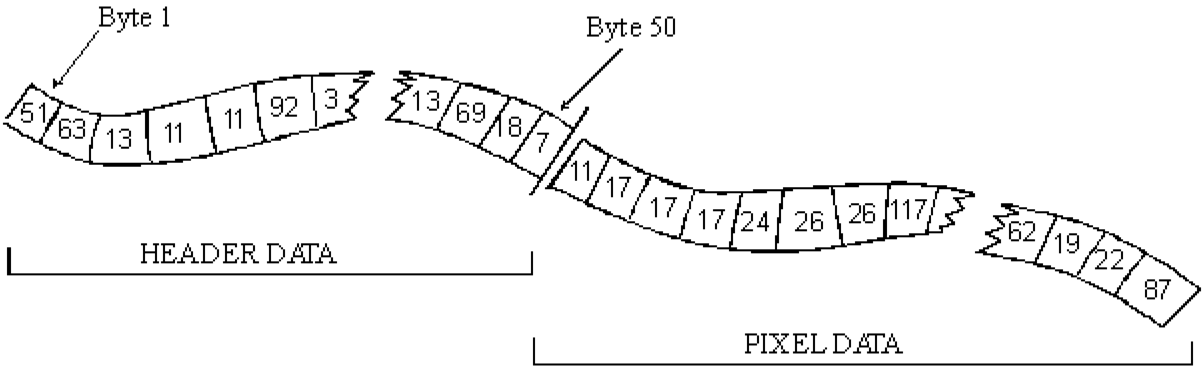
(1)

(b) Images can be saved in a bitmapped image file as a **'256 colour bitmap'**.

How many bytes are used to store each pixel? ______________________________

(1)

(c) The first 50 bytes of these bitmapped files are used for **header data**. See **Figure 1**.



Name **two** items of data which should be included and stored in the file header.

1. ____

2. ____

(2)

(d) A high level programming language has a function `ReadImageByte` which is used to read the contents of a bitmapped image file.

It is defined in the help files as follows:

Function `ReadImageByte : Byte` The function `ReadImageByte` returns the next byte of data from a bitmapped image.

The pseudo-code that describes the process of reading the contents of the file header data is shown below.

```
Procedure ReadHeaderData
   For Position ← 1 To 50 Do
     CurrentHeader [Position] ← ReadImageByte
   EndFor
EndProcedure
```

(i) Complete the identifier information in the table below for this pseudo-code.

Variable Identifier	Data Type	Description
`Position`	Integer	
`Current Header`		Stores theheader data

(2)

The first four bytes of the header data are:

First	Second	Third	Fourth
51	63	13	11

(ii) What **binary value** will be assigned to variable `CurrentHeader[3]`?

____

(1)

(e) The width and height of the bitmapped image are stored by variables `ThisWidth` and `ThisHeight`.

A procedure `ReadPixelData` is to read the remaining contents of a bitmap image i.e. the byteswhich represent the individual pixels and to organise these as an image grid as shown in Figure 2.

Byte 51

Byte 58

Byte 59

11	17	17	17	24	26	26	117
19	50	25	96	96	24	24	113
18	114	22	87	13	29	31	45
81	96	28	87	29	49	45	45
39	101	28	28	62	19	22	87

Byte 98

(i) Complete the gaps in the pseudo-code below.

```
Procedure ReadPixelData
  For X ← 1 To ThisHeight Do
    For Y ← 1 To ____________ Do
      ThisByte ← ReadImageByte
      ByteData [ ______________, Y] ← ThisByte
    EndFor
  EndFor
EndProcedure
```

(2)

(ii) What data structure has the programmer used for variable `ByteData`?

____

(1)

(f) A graphics studio has produced all the graphic images for a new computing textbook.

The images all need to be ‘tidied up’ and, rather than edit every one with graphics software, it is suggested that the task be given to a computer programmer who will, for each image:

- remove the top row of pixels, and

- remove all the pixels in the first two columns - see **Figure 3**.

Byte 51 Byte 58

Byte 59

255	255	255	255	255	255	255	255
255	25	25	96	96	24	24	113
255	114	22	87	13	29	31	45
255	96	28	87	29	49	45	45
255	101	28	28	27	71	23	23

Byte 98

The `ReadPixelData` procedure is to be refined so that not all pixels will be retained. **The enclosed pixels in Figure 3 are those to be retained** and these bytes will be written to an array `Final`. These pixels, together with the header data bytes, will form the amended bitmapped file.

The test pixel data shown in **Figure 3** are to be used to trace the amended `ReadPixelData` procedure.

```
ThisWidth = 8
ThisHeight = 5
Call ReadPixelData

Procedure ReadPixelData
  Counter ← 0
  For X ← 1 to ThisHeight Do
    For Y ← 1 to ThisWidth Do
      ThisByte ← ReadImageByte
      If(X>1 AND Y>2)Then
        Final [Counter] ← ThisByte
        Counter ← Counter + 1
      EndIf
    EndFor
  EndFor
EndProcedure
```


Trace the execution of the pseudo-code **for two iterations only** of the outer loop (the loop controlled by variable X) by completing **Figure 4**.

ThisWidth	ThisHeight	Counter	X	Y	This Byte		Final
8	5					[0]	
						[1]	
						[2]	
						[3]	
						[4]	
						[5]	
						[6]	
						[7]	
						[8]	
						[9]	
						[10]	
						[11]	
						[12]	
						[13]	
						[14]	
						[15]	

(6)

(g) In this question identifier names have been used in the design for variables and procedure and function names.

(i) Name **one** other program element for which the programmer would allocate an identifier name.

____

(1)

(ii) Programming languages impose restrictions about the choice of identifier names; for example a <Space> character cannot be included.

State **two** other restrictions in a programming language with which you are familiar.

____

____

(2)

(Total 20 marks)

Q21.

Cars over three years old have to pass a roadworthy test called the MOT. Various categories are tested and for this question they have been simplified to:

- Brakes
- Steering
- Tyres
- Bodywork.

A car passes the MOT test – in this simplified scenario – if it passes all four categories.

Data for a single car is stored as a string consisting of the digit characters ‘0’ and ‘1’ e.g. ‘1110’.

- ‘1’ denotes a category pass
- ‘0’ denotes a category fail.

The order of the categories is as shown above. For example, the data ‘1110’ describes a car which passed on brakes, steering and tyres, but failed on bodywork.

The built-in function `SingleCharacter` is to be used in the algorithm which follows, and is described in the help files as follows:

`SingleCharacter(ThisString: String; ThisPosition : Integer) : Char ;`
Returns the single character at position `ThisPosition` in the string `ThisString`.
E.g. `Result :=SingleCharacter('1110', 4)` would return and assign ‘0’ to `Result`

The following incomplete algorithm is designed to calculate whether a single car has passed or failed.

The identifier list for variables used by the algorithm is shown in **Table 1**.

(a) Complete **A**, **B** and **C** in the algorithm.

```
CarFailed ← False
Input NextCar
For Position ← 1 To 4
  Do NextCategory ← SingleCharacter   ( A ____________, B ____________ )

      If C ______________________________________________
        Then CarFailed ← True
      End If
End For


If CarFailed = False
   Then Output 'Car passed MOT'
   Else Output 'Car failed MOT'
```

```
End If
```

(3)

(b) Complete the data types and comment – **D**, **E** and **F** – in **Table 1**.

The data types should be selected from those shown in **Table 2**.

Table 1

Variable	Data Type	Comment
`Position`	**D**____________________	**E**____________________
`NextCar`	String	Data for a single car
`NextCategory`	**F**____________________	Data for a single category
`CarFailed`	Boolean	Result indicator

(3)

Table 2

Data type	Explanation
Integer	Whole number
Real	Number witha fractional part
String	Zero or morecharacters
Char	Singlecharacter
Boolean	True/Falsevalues only

(Total 6 marks)

Q22.

A firm selling double glazing employs three sales staff. Each person is given a sales target for each of the four quarters of the year.

- Quarter 1 January – March
- Quarter 2 April – June
- Quarter 3 July – September
- Quarter 4 October – December

Based on all the sales made, the data in **Table 1** is produced showing whether or not each sales person achieved their target sales for each quarter. Each value is stored as a single character ‘Y’ (sales target met) or ‘N’ (sales target not met).

The columns represent each quarter, each row represents a salesperson.

Table 1

		Target		
	[1]	[2]	[3]	[4]
[1]	Y	N	Y	N
[2]	N	N	**Y**	Y
[3]	N	N	N	N

(a) What data structure could be used in a programming language for organising the data shown in **Table 1**?

____

(1)

(b) One of the data values in **Table 1** has been emboldened. What does this value represent?

____

(1)

(c) The following algorithm processes the data shown in **Table 1**. Trace the execution of the algorithm by completing **Table 2**.

```
For Quarter ← 1 To 4
   Do NewArray [Quarter] ← 0
End For
For Person ← 1 To 3
    Do
     For Quarter ← 1 To 4
         Do
           If Target[Person, Quarter] = 'N'
           Then NewArray [Quarter] ← NewArray [Quarter] + 1
         End If
     End For
End For
```


Table 2

Person	Quarter	Target[Person, Quarter]	NewArray [1] &nb	sp;[2]	[3]	[4]
	1					

(6)

(d) Explain what numbers are being calculated and stored in the `NewArray` data structure.

____

____

(2)

(Total 10 marks)

Q23.

A county has a number of local libraries in various towns. Books currently belong to each library and there is no system for the exchange of books between libraries.

New programs have to be written, as the decision has been made to have centralised records of library books.

The software house commissioned to write the new programs has obtained a complete list of titles held at each library. It found that a common system was used for the book codes. Some older books will not be retained and this is to be indicated by the ToBeRetained column in the table below.

BookTitle	BookCode	YearFirstInStock	ToBeRetained
Hang-gliding made simple	T05320	1993	
Around the world in 80 days	T76542	2001	
My way	M11981	1990	
Starting with hypnotherapy	M79080	2005	
Kim Smith – the autobiography	M00876	1991	
XXX			

(a) Study the sample data shown in the table. This data will be accessed by program code. Name the most suitable **data type** which should be used for each data item. Each data type **must be different.**

(i) BookCode ____

(1)

(ii) YearFirstInStock ____

(1)

(iii) ToBeRetained ____

(1)

(b) The first application to be developed is a program to search the complete list of books and to calculate the data values for the ToBeRetained column; any books which were bought before 1992 will not be retained.

The incomplete pseudo-code which follows shows a first attempt at the algorithm. Data for each of the four attributes BookTitle, BookCode, YearFirstInStock, ToBeRetained are shown in the table above, and are to be stored in four arrays BookTitle, BookCode, YearFirstInStock and ToBeRetained.

Complete the pseudo-code in the **three** places indicated.

```
For Book ← 1 To TotalNoOfBooks
    If YearFirstInStock[ (i) ___________________ ] < 1992

       Then ToBeRetained[Book] ← (ii) ___________________

       Else ToBeRetained[Book] ← (iii) ___________________
    EndIf
EndFor
```

(3)

(c) A second program is to be developed to allocate each book a new code number. The old book codes are to be abandoned. The first character of the old book code indicates the book's location.

- This book location is to be retained and stored in an array Location.
- Each new book code will be a unique integer number that will be generated by the program. The first number will be 1.

Use will be made of a 'built-in' function StartString. It is defined in the help files as follows:

Function StartString(ThisString : **String;** NoOfCharactersToRetain **: Integer) :String ;** **The function is given the string** ThisString **and returns the number of characters specified by** NoOfCharactersToRetain **starting from the first character of** ThisString.

(i) What are the values of the **parameters** used in the following code?

```
NewString : = StartString('T76542', 1)
```

1. ____

2. ____

(2)

(ii) What value is assigned to `NewString` when this code is executed?

____

(1)

(iii) The pseudo-code for the algorithm to calculate the new book codes and the locations is shown below.

```
NextAvailableCode ← 1

Book ← 1
Repeat
     If YearFirstInStock[Book] >=1992
       Then
         Begin

           LocationLetter ← StartString(BookCode[Book], 1)
           If LocationLetter = 'T'

             Then Location[Book] ← 'Torrington'
           If LocationLetter = 'M'

              Then Location[Book] ← 'Morristown'

            NewCode[Book] ← NextAvailableCode

            NextAvailableCode ← NextAvailableCode + 1
         End

     Book ← Book + 1
Until BookTitle[Book] = 'XXX'
```

Trace the execution of this algorithm by completing the trace table **Figure 2**; use the data shown in the table **Figure 1.**

Show also the final contents of the `Location` and `NewCode` arrays in **Figure 3** and **Figure 4.**

Figure 1

	BookTitle		BookCode		YearFirstInStock
[1]	Hang-gliding made simple	[1]	T05320	[1]	1993
[2]	Around the world in 80 days	[2]	T76542	[2]	2001
[3]	My way	[3]	M11981	[3]	1990
[4]	Starting with hypnotherapy	[4]	M79080	[4]	2005
[5]	Kim Smith – the autobiography	[5]	M00876	[5]	1991
[6]	XXX	[6]		[6]	

Figure 2

NextAvailableCode	Book	LocationLetter

1	1	‘T’

Figure 3 Location

[1]	
[2]	
[3]	
[4]	
[5]	

Figure 4 New Code

[1]	
[2]	
[3]	
[4]	
[5]	

(6)

(Total 15 marks)

Q24.

A *recursively-defined* procedure **ProcA** that takes two integers as parameters is defined below.

(a) What is meant by a recursively-defined procedure?

___

___

(1)

(b) What is the role of the stack when a recursively-defined procedure is executed?

___

___

(1)

(c) Dry run the procedure call **ProcA(11,1)** using the data in the array, **Items**, by completing the trace table below.

```
Procedure ProcA (Number,Entry)
      If Number <> Items[Entry]
      Then ProcA (Number,Entry+1)
```

Items

[1]	4
[2]	5
[3]	8

```
        Else Output (Entry)
        EndIf
    EndProc
```

[4]	11
[5]	15
[6]	19
[7]	21
[8]	28
[9]	33

Number	Entry	Output
11	1	

(4)

(d) What is the purpose of this algorithm?

____

(1)

(e) Give a situation where this algorithm will fail.

____

____

(1)

(f) Suggest a modification to the algorithm that will prevent it from failing.

____

____

(1)

(g) With an ordered array, Items, of many more entries, what more efficient algorithm could be used to achieve your expressed purpose in part (d)?

____

____

(1)
(Total 10 marks)

Q25.

(a) Well constructed programs use a structured approach for the design and coding stages.

One practical way in which the programmer will use a structured approach to programming is the use of subroutines (procedures/functions). Give **three** other ways.

1. ____

2. ____

3. ____

(3)

(b) A program is to be written which calculates the hourly pay rate for an employee. The calculation is based on the number of complete years the employee has worked for the firm (e.g. 3 years). All employees get a basic £7.88 per hour. For each year worked, up to a maximum of 5 years only, an additional £0.65 is added to the basic hourly rate.

The algorithm for this program is as follows:

1. Enter the surname
2. Enter the number of years of service
3. Calculate the employee's pay rate
4. Output the surname and pay rate

(i) Complete the table showing **three** variable identifiers and their data types you would use for this problem.

Variable Identifier	Data Type

(3)

(ii) The detail for step 3 in the algorithm is broken down into more detail as follows:

3.1 If the number of years of service value is over 5, then change the value stored to 5

3.2 Calculate the employee's pay rate

Write pseudo-code for these two steps using the appropriate identifiers from the table.

3.1 ____

____

3.2 ____

____

(3)

(Total 9 marks)

Q26.

A company makes sofas and operates seven days a week. Each day a record is made of the number of sofas that are rejected at the final quality control stage. An average of one reject each day is considered acceptable. This is investigated using the program below at the end of each week.

```
Program RejectReport;
Var
     DayNo: Integer;
     RejectTotal: Integer;
     DailyRejects: Array [1..7] of Integer;
Begin
     RejectTotal := 0;
     For DayNo := 1 To 7
          Do RejectTotal := RejectTotal + DailyRejects [DayNo];
     WriteLn(RejectTotal);
End.
```

(a) What does this program do?

____

____

____

(2)

(b) (i) Write the assignment statement in the program which performs a calculation.

____

(1)

(ii) Write a declaration statement that appears in the program.

____

(1)

(iii) What is the purpose of the variable `DayNo`?

____

(1)

(iv) What type of data structure is `DailyRejects`?

____

(1)

(c) The program is to be extended to report whether this was a satisfactory week for the number of rejected sofas. An average of one reject each day is considered acceptable.

Write additional programming statement(s), in the language you are familiar with, to report one of the messages ‘Investigate’ or ‘Inside weekly tolerance’. Use the same variable identifiers as used in the program given.

____

____

____

____

____

____

(2)

(d) "*A programming team should make extensive use of program libraries*."

Explain this statement ____

____

____

____

(2)

(e) Another application is to be developed. The number of rejects per week is recorded over a five-week period. This data is stored in array `NoOfRejects`. The array `WeeklySupervisor` records who the supervisor was for week 1, week 2, etc. A third array `SupervisorTotal` will record the total number of unsatisfactory weeks for each of the three supervisors.

The pseudo-code which follows in **Figure 1** makes clear which array position is used for each supervisor.

Figure 1

	`NoOfRejects`		`WeeklySupervisor`		`SupervisorTotal`
[5]	9	[5]	'Jones'		
[4]	8	[4]	'Summers'		
[3]	1	[3]	'Jones'	[3]	
[2]	9	[2]	'Summers'	[2]	
[1]	8	[1]	'Franks'	[1]	

```
SupervisorTotal [1] ←0

SupervisorTotal [2] ←0

SupervisorTotal [3] ←0

For WeekNo ←1 to 5

     ThisNumber←NoOfRejects [WeekNo]
     If ThisNumber > 7 Then
          Output 'Investigate'
          Call AddToSupervisorTotal
     End If
End For
Procedure AddToSupervisorTotal
If WeeklySupervisor [WeekNo] = 'Franks'
```

```
    Then SupervisorTotal [1] ←SupervisorTotal [1] + 1
End If
If WeeklySupervisor [WeekNo] = 'Summers'

    Then SupervisorTotal [2] ←SupervisorTotal [2] + 1
End If
If WeeklySupervisor [WeekNo] = 'Jones'

    Then SupervisorTotal [3] ←SupervisorTotal [3] + 1
End If
End Procedure
```

(i) The number of unsatisfactory weeks when Jones was in charge is stored in the array `SupervisorTotal`. At what position in the array is this number stored?

____

(1)

(ii) Trace the algorithm by completing the trace table in **Table 1.**

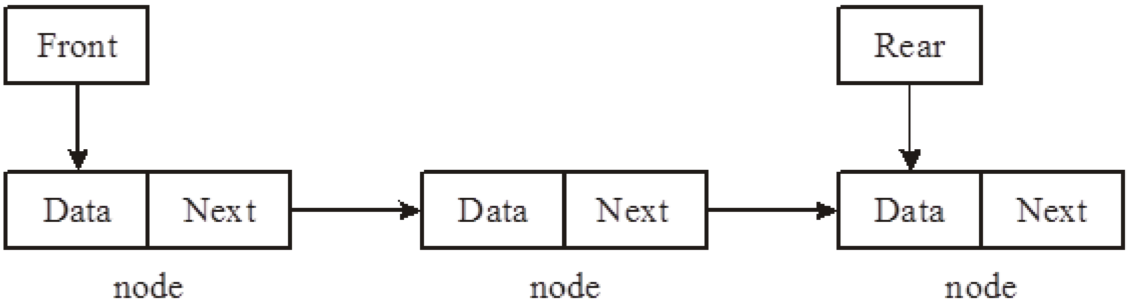
Table 1

WeekNo	ThisNumber	Output	SupervisorTotal [1]	[2]	[3]
			0	0	0
1					

(6)

(Total 17 marks)

Q27.



(a) Assume a queue is implemented as a linked list using pointers as in the figure.

Give the **three** steps required to remove a node from the front of the queue and recover the memory space occupied by the node.

1. ____

2. ____

3. ____

(3)

(b) A set of operations are defined to manipulate the contents of the queue. As well as *Remove* these include *FrontItem* and *IsQueueEmpty*.

Name another operation that would be essential to use this queue.

____

(1)

(c) The queue could be implemented using an array instead of a linked list.

(i) What additional operation will be required if the queue is implemented using an array?

____

(1)

(ii) Give **one** advantage of array implementation.

____

(1)

(iii) Give **two** disadvantages of array implementation.

1. ____

2. ____

(2)

(Total 8 marks)

Q28.

An integer array A contains the following items.

	A
[1]	**3**
[2]	**5**
[3]	**11**
[4]	**12**
[5]	**18**

[6]	21
[7]	26
[8]	29
[9]	32

The operator DIV performs integer division. x DIV y calculates how many times y divides exactly into x. For example 7 DIV 3 = 2.

(a) Dry run the following algorithm by completing the trace table.

```
Number ← 12
Lower ← 1
Upper ← 9
While Lower<Upper
     Current ← (Lower+Upper)DIV 2
     If Number >= A [Current] Then Lower ← Current
     If Number <= A [Current] Then Upper ← Current
EndWhile
Return Current
```

Number	Lower	Upper	Current

Value returned	

(8)

(b) What is the purpose of this algorithm?

____

(1)

(Total 9 marks)

Q29.

A retail store employs ten sales staff. Staff try to persuade customers to take out a store card with the company when they make a purchase. The store keeps a record of the number of new store cards issued by its sales staff over the first six months of the year.

Table 1

StoreCards

	[1]	[2]	[3]	[4]	[5]	[6]
[1]	12	12	6	8	3	2
[2]	12	17	7	4	5	6
[3]	2	12	0	12		
[4]	4	10	7	4		
[5]	5	0	0	0	0	0
[6]	6	1	4	6	7	8
[7]	12	19	12	**16**	17	6
[8]	13	9	7	3	4	5
[9]	12	8	4	4	5	4
[10]	14	11	12	4	5	6

The data is to be stored in a 2-dimensional array with identifier StoreCards as shown in the table above The first subscript of the array represents the row number (the salesperson number), and the second subscript the column number (the month).

(a) In the table the value 16 has been **emboldened**. Explain what this value represents.

____ **(2)**

(b) Write a declaration statement for the array StoreCards.

____ **(2)**

(c) Using the data given in the table abobe, write an assignment statement for the January sales for salesperson 8.

____ **(2)**

(d) Study the pseudo-code below.

```
Input SalesPersonNumber
PersonTotal ← 0
For Month ← 1 to 6 Do
     PersonTotal ← PersonTotal +
```

```
        storeCards[SalesPersonNumber, Month]
    End For
    Print PersonTotal
```

Explain what this algorithm is designed to do.

____

____

____

____

(2)

(e) A number of programs are to be written for the store card application, and the following are some of the data values which will need to be stored and/or calculated.

State what data type the programmer would use for each data item below.

(i) Average overtime hours worked by each member of staff.

____

(1)

(ii) Whether or not the staff are willing to work on Boxing Day.

____

(1)

(iii) The number of customer complaints made about each member of staff.

____

(1)

(Total 11 marks)

Q30.

Table 1

ASCII Code Table

Character	Decimal	Character	Decimal	Character	Decimal
<Space>	32	I	73	R	82
A	65	J	74	S	83
B	66	K	75	T	84
C	67	L	76	U	85
D	68	M	77	V	86

E	69	N	78	W	87
F	70	O	79	X	88
G	71	P	80	Y	89
H	72	Q	81	Z	90

(a) Use the ASCII code table given in **Table 1** to look up the ASCII code for character 'V'

(i) What is its representation when written in 7-bit binary?

(1)

(ii) What is its value when expressed in 8 bits with the 8th bit an odd parity bit?

(1)

(b) A programming language help file describes the Chr( ) function as follows.

> Chr( ) takes a single integer value as its parameter.
>
> The function returns the ASCII character represented by the parameter.
>
> Example: Chr(65) will return value 'A'.

(i) What is returned by Chr(68)?

____

(1)

(ii) What value is assigned to variable MyChar when the following **two** statements are executed?

```
Value ← 9

MyChar ← Chr (65   +   Value)
MyChar = ____________________________________________
```

(1)

(c) The algorithm which follows uses a function ConCat.

> The ConCat function takes **two** strings as its parameters, and returns the concatenated string.
>
> Example: ConCat('Fred', 'Smith') would return 'FredSmith'.

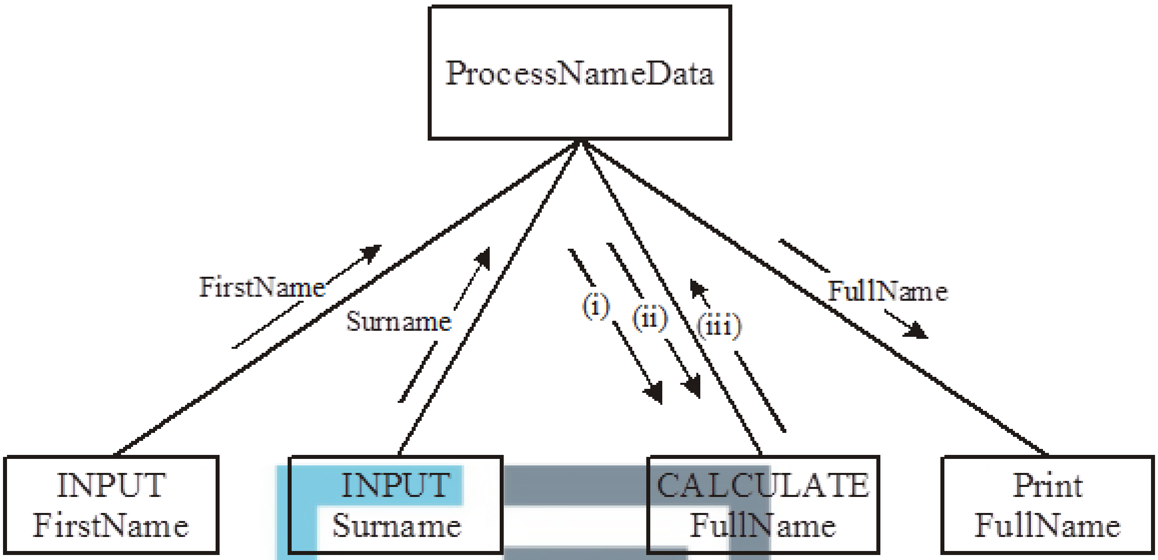
```
Procedure
    ProcessNameData
```

```
    INPUT  FirstName
    INPUT  Surname

    FullName ← ConCat (FirstName, Surname)

    PRINT  FullName
End Proc
```

The stages of this procedure ProcessNameData are shown as a structure chart below.



What are the missing labels?

(i) ____ **(1)**

(ii) ____ **(1)**

(iii) ____ **(1)**

(d) **Table 2** shows an array of integers with identifier Index, to which values have been assigned.

Table 2

Index

[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
0	13	–33	4	17	17	14	17

Study the following algorithm and trace its execution by completing the trace table **Table 3**, using the ASCII code table given in **Table 1**.

```
 Final String ← '   '

For Position ←1 To 8 Do

     NextNumber ← 65 + Index[Position]
```

```
        NextChar ← Chr(NextNumber)

        FinalString ← ConCat(FinalString, NextChar)
    End For
    Print FinalString
```

Table 3

Position	NextNumber	NextChar	FinalString
1	65	'A'	'A'
2			

(6)

(Total 13 marks)

Q31.

A *linear search* and a *binary search* are **two** different methods of searching an ordered list. A given list contains 271 items.

(a) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a linear search?

____

(1)

(ii) Explain your answer.

____

____

(1)

(b) (i) What is the maximum number of items accessed when searching for a particular item from the given list using a binary search?

____

(1)

(ii) Explain your answer.

(1)

(c) An integer array A contains the following items.

A	
[1]	23
[2]	45
[3]	16
[4]	12
[5]	31

(i) Dry run the following algorithm by completing the trace table.

```
For Count1 ← 1 To 4
   For Count2 ← 1 To 4
      If A[Count2] > A[Count2 + 1] Then
     Temp ← A [ Count2]
     A[Count2] ← A[Count2 + 1]
     A[Count2 + 1] ← Temp
      EndIf
   EndFor
EndFor
```

Count 1	Count2	Temp	A				
			[1]	[2]	[3]	[4]	[5]
–	–	–	23	45	16	12	31
1	1						

(5)

(ii) What is the purpose of this algorithm?

(1)

(iii) Suggest **one** way the algorithm could be improved.

(1)

(Total 11 marks)

Q32.

(a) (i) Explain **one** difference between a procedure and a function.

(2)

(ii) Name and describe a built-in function you have used in your programming work, or when using a generic software package.

(2)

(b) A particular built-in function is described in a programming language's help files as follows:

> Function MatchString(ThisString , StringSearchedFor : String) :Boolean
>
> The function **MatchString** returns a **Boolean** value indicating whether or not the string **StringSearchedFor** appears within the string **ThisString**.

An error is returned when a function call is incorrectly formed.

What value is returned to the Result1, Result2 and Result3 variables from the following function calls?

(i) Result1 := MatchString (‘Harry Potter’, ‘Pot’)

____

(1)

(ii) Result2 := MatchString (‘Potter’, ‘Harry Potter’)

____

(1)

(iii) Result3 := MatchString (‘Harry Potter’, 59)

____

(1)

(c) In part (b) (i) Result1 is an identifier used for a variable. Name **two** other uses for identifiers in a high level language.

1. ____

2. ____

(2)

(d) The programming language being used has both compiler and interpreter software for program development.

Give **one** advantage of the use of each.

Interpreter advantage ____

____

____

(1)

Compiler advantage ____

____

____

(1)

(Total 11 marks)

Q33.

The data shown below is a list of surnames of 20 motor car policyholders with the number of claims they have each made in the last five years.

PolicyHolder	NoOfClaims

1	Wilcox	1	1
2	Adams	2	0
3	Pollard	3	0
4	Williams	4	0
5	Searle	5	3
6	Kelly	6	0
7	Lewis	7	1
8	Franks	8	5
9	Patel	9	1
10	Li Che	10	0
...		...	...
...		...	...
19	Wilkinson	19	3
20	Veale	20	0

(a) (i) The user inputs a policyholder. If the surname is found, the program outputs the number of claims for that policyholder.

```
          Read(SearchName)
          For P := 1 To 20 Do
                   If PolicyHolder[P] = SearchName
                       Then GoTo 200
          GoTo 300
    200 : Write(NoOfClaims[P])
    300: End
```

Give **two** reasons why this is badly designed program code.

1. ____

____

2. ____

____

(2)

(ii) Write declaration statements (in a language with which you are familiar) for the PolicyHolder or NoOfClaims data structure above, and one other variable used in the code above.

The programming language I am using is ______________________________

1. ____

2. ____

(2)

(b) A new task is to design and write code to establish if there are any policyholders who have made five or more claims. The program will output a 'yes' or 'no' message only.

Write the code for this new task in a programming language with which you are familiar.

(Hint: Use a loop structure to initiate the loop, and then end the loop when some condition is met.)

____

____

____

____

____

____

____

____

____

____

____

____

____

____

(5)

(Total 9 marks)

Q34.

A tree has the following functions defined:

RootValue(T)	Returns the contents of the root node of the tree T
LeftChild(T)	Returns the left child of the root node of the tree T
RightChild(T)	Returns the right child of the root node of the tree T

A recursively-defined procedure P with a tree as a parameter is defined below.

```
Procedure P (T)
     If LeftChild(T) exists
          then P(LeftChild(T))
     Output RootValue(T)
     If RightChild(T) exists
```

```
            then P(RightChild(T))
    EndProc
```

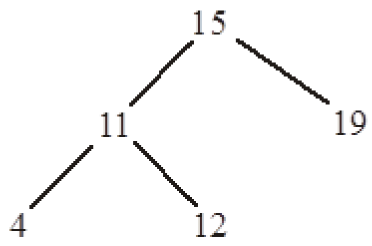
(a) What is meant by recursively-defined?

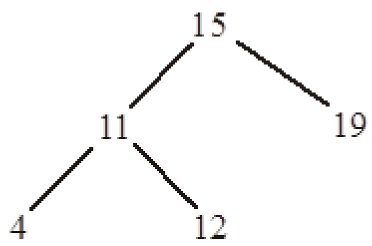
____

____

(1)

(b) (i) Complete the table below by dry running the procedure call P(T) for the tree T given below.



Procedure Call	T		Output
P_1			

(6)

(ii) What does procedure P describe?

____

(2)

(Total 9 marks)

Q35.

The following algorithm uses an array Values that contains the integers 4,7,9.

(a) Dry run this algorithm by using the trace table below.

```
Last ← 3
New ← 6
Ptr ← 1
WHILE(New > Values[Ptr])
  Ptr ← Ptr + 1
ENDWHILE
WHILE (Last >= Ptr)
  Values[Last+1] ← Values[Last]
  Last ← Last - 1
ENDWHILE
Values[Ptr] ← New
```

New	Last	Ptr	Values

			[1]	[2]	[3]	[4]	[5]
6	3	1	4	7	9		

(6)

(b) What is the purpose of this algorithm?

____

____

(1)
(Total 7 marks)