1(a) The temperatures of an ocean are input into a computer system. They are recorded, and will be accessed, in the order in which they arrive. The data for one week is shown:

5, 5.5, 5, 6, 7, 6.5, 6

The data is to be stored in a data structure. The programmer stores the data in a queue.

Explain why a queue is used instead of a stack.

---

[2]

(b) The data is processed. After processing, the value for the first day is stored as 0. The value for each following day is stored as an increase, or decrease, from the first day.

For example: if the first day was 7, the second was 6 and the third was 9, after processing it would be stored as 0, –1, 2.

i. The queue uses `dequeue()` to return the first element of the queue.

`dequeue()` is a function.

Explain why `dequeue()` is a function, not a procedure.

---

[1]

---

ii. Complete the algorithm to process the data in the queue and store the results in an array called `processedData`.

```
processedData[0] = 0

firstDay = ..........................

for count = 1 to 6

   processedData[.......................] = dequeue() – .......................
```

```
next count
```

[3]

iii. The contents of `processedData` are shown.

0	0.5	0	1	2	1.5	1

The data needs to be sorted into ascending order.

Explain how a bubble sort algorithm sorts data. Use the current contents of `processedData` in your explanation.

[5]

iv. A bubble sort has the following complexities:

Best time	O(n)
Average and worst time	O(n^2)
Worst space	O(1)

Describe what each of these complexities mean.

Best time O(n)

Average and worst time O(n^2)

Worst Space O(1)

[6]

2(a) A program needs to store the names of plants that are in a garden, so they can be easily found and accessed in alphabetical order.

The data is stored in a tree structure. Part of the tree is shown.

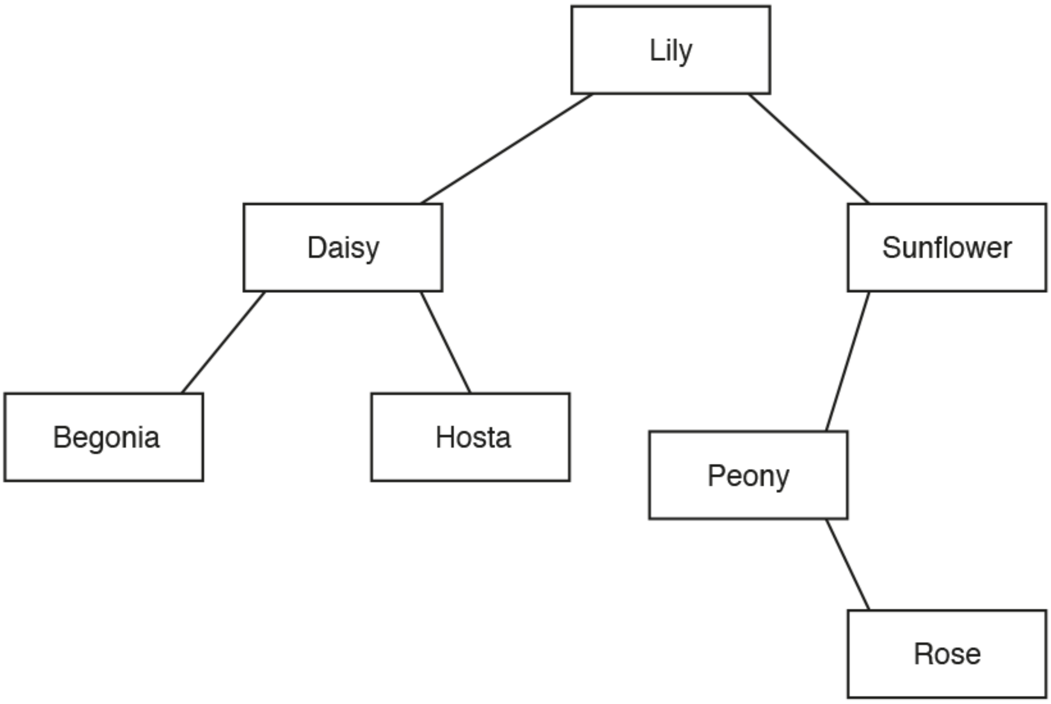


Fig. 2.1

i. State the type of tree shown in Fig. 2.1.

____ **[1]**

ii. Show the output of a breadth-first traversal of the tree shown in Fig. 2.1.

____

____

____

____

____

____

____

[3]

iii. Explain how backtracking is used in a depth-first (post-order) traversal. Use the tree in Fig. 2.1 in your explanation.

[4]

(b) The elements in the tree in Fig. 2.1 are read into a linked list producing an alphabetised list.

i. Complete the following table to show the linked list for the data.

Data item	Data	NextPointer
0	Begonia	
1	Daisy	
2	Hosta	
3	Lily	
4	Peony	
5	Rose	
6	Sunflower	

[2]

ii. A new plant, Lavender, needs adding to the linked list. The linked list needs to retain its alphabetical order.

Complete the table to show the linked list after Lavender is added.

Data item	Data	NextPointer
0	Begonia	
1	Daisy	
2	Hosta	
3	Lily	
4	Peony	
5	Rose	
6	Sunflower	

[3]

iii. Hosta needs removing from the linked list.

Explain how a data item is removed from a linked list. Use the removal of Hosta in your answer.

---

[4]

iv. The linked list is stored as a 2D array with the identifier `plantList`. The index of the first element of the linked list is stored in the identifier `firstElement`.

All contents of the linked list need to be output in alphabetical order.

Write an algorithm to follow the pointers to output the contents of the linked list in alphabetical order.

Add comments to explain your code.

[5]

3(a) A recursive function, GCD, is given in pseudocode.

```
function GCD (num1, num2)

        if num2 == 0 then

                return num1

        else

                return GCD(num2, num1 MOD num2)

        endif

endfunction
```

The function uses branching.

i. Identify the type of branching statement used in the function.

____ [1]

ii. Explain the difference between branching and iteration.

____

____

____

____ [2]

iii. Identify the **two** parameters in the function.

1 ____

2 ____

[1]

iv. State whether the parameters should be passed by value, or by reference. Justify your answer.

____

____

[2]

v. Describe the arithmetic operation of `MOD`. Use an example in your answer.

[2]

(b) Trace the recursive function when it is called by the statement `GCD(250, 20)`. Give the final value returned.

Final return value:

[3]

(c) The function has been rewritten using iteration instead of recursion.

i. State **one** benefit and **one** drawback of using iteration instead of recursion.

Benefit --

Drawback --

----

[2]

ii. Complete the missing statements in this iterative version of the function.

```
function newGCD (num1, num2)
   temp = 0
   while (num2 != ..................)
     .................. = num2
     num2 = num1 MOD ..................
     num1 = temp
   endwhile
   return ..................
endfunction
```

[4]

4(a) Mabel is a software engineer. She is writing a computer game for a client. In the game the main character has to avoid their enemies. This becomes more difficult as the levels of the game increase.

Mabel uses decomposition to design the program.

Explain how decomposition can aid the design of this program.

--- [2]

(b) The computer game allows a user to select a character (e.g. name, gender). They can then choose a level for the game (easy, normal, challenging). The user controls their character by moving it left or right. The character can jump using space bar as an input. If the character touches one of the enemies then it loses a life. The character has to make it to the end of the level without losing all their lives.

The game is designed in a modular way.

i. One sub-procedure will handle the user input.

Describe **three** other sub-procedures Mabel could create for the given game description.

1 ---

---

2 ---

---

3 ---

----

[6]

ii. Describe the decision that the program will need to make within the user input subprocedure and the result of this decision.

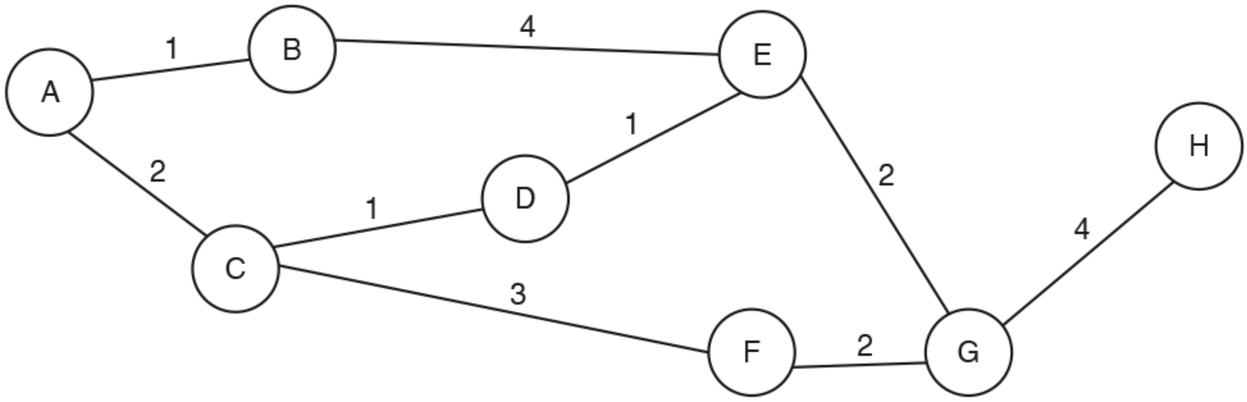
-- [2]

iii. Define pipelining and give an example of how it could be applied in the program.

-- [2]

(c) The game's 'challenging' level has intelligent enemies that hunt down the character in an attempt to stop the user from winning. The program plans the enemies' moves in advance to identify the most efficient way to stop the user from winning the game.

The possible moves are shown in a graph. Each node represents a different state in the game. The lines represent the number of moves it will take to get to that state.



Show how Dijkstra’s algorithm would find the shortest path from A to H.

[6]

(d) * Mabel has been told that true programmers write programs in a text editor, and do not use IDEs. Mabel does not agree with this statement.

Discuss the use of an IDE in the development of this program.

[9]

5(a) A 1-dimensional array stores the following data:

Index	0	1	2	3	4	5
Data	2	18	6	4	12	3

The array needs sorting into descending order.

Describe how a merge sort would sort the given array into descending order.

[6]

(b) An insertion sort can be used to sort the array instead of a merge sort.

Explain why an insertion sort might use less memory than a merge sort.

____

____

____

____ **[2]**

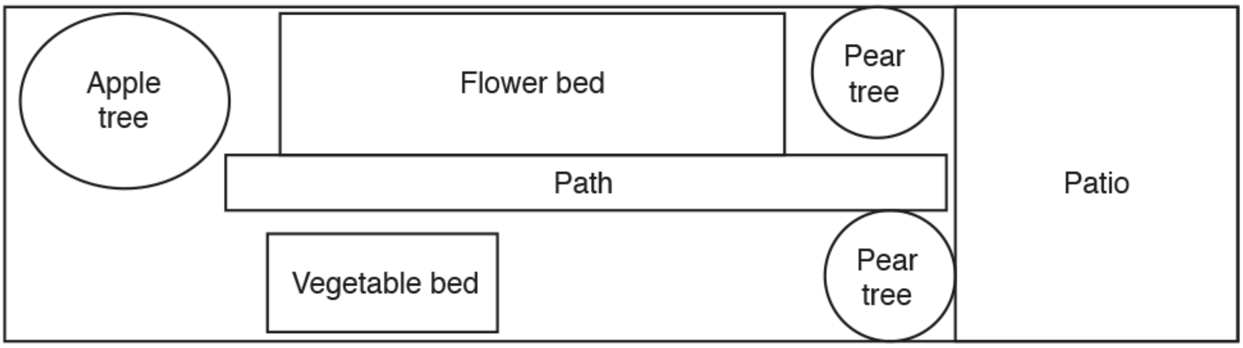
6 * Benedict runs a social networking website. He has been told he should use data mining to help him enhance and improve his website.

Evaluate the use of data mining to help Benedict enhance and improve his social networking website.

[9]

7(a) A program is needed to plan the layout of a garden.

The program will allow the user to create an image of the garden, for example:



The program is to be built using object oriented programming.

All items that can be added to the garden are declared as instances of the class `GardenItem`.

The class has the following attributes:

Attribute	Description	Example
`itemName`	The name of the item	Flowerbed
`length`	The length of the item in metres	2
`width`	The width of the item in metres	1

i. The constructor method sets the attributes to values that are passed as parameters.

Write pseudocode or program code to declare the class `GardenItem` and its constructor. All attributes should be private and initialised through the constructor (e.g. `daisies = new GardenItem("Flowerbed",2,1)`).

[4]

ii. The trees in the garden layouts are defined by the class `Tree`. This class inherits from `GardenItem`.

The class `Tree` has the additional attributes: `height, sun, shade`.

If `sun` is `true` then the tree can grow in full sun, if it is `false` then it cannot.

If `shade` is `true` then the tree can grow in full shade, if it is `false` then it cannot.

The length and width of a tree are the same. Only one value for these measurements is passed to the constructor.

Write an algorithm, using pseudocode or program code, to declare the class `Tree`.
Declare all attributes as private.

[5]

iii. The Common Oak is a type of tree. It has a maximum height, length and width of 40 m. It can grow in sun and shade.

Write a statement, using pseudocode or program code, to declare an instance of tree for the Common Oak. Give the object the identifier `firstTree`.

[4]

iv. The classes `GardenItem` and `Tree` use get and set methods to access and alter their private attributes.

Write the get method `getItemName` and set method `setItemName` for class `GardenItem`. The set method takes the new value as a parameter.

Do not write any other methods, or re-declare the class.

[4]

v. The trees in the garden layouts are stored in a 1-dimensional array, `treeArray`. The array can store a maximum of 1000 items. The array has global scope.

A procedure, `findTree`, takes as parameters:

- The maximum height of a tree
- The maximum width of a tree
- Whether the tree can live in full sun
- Whether the tree can live in full shade.

It searches the array, `treeArray`, for all trees that do not exceed the maximum height and width, and that can grow in the conditions available. If there are no suitable trees, a suitable message is output.

It outputs the name and details of the trees found in an appropriate message.

Call the get methods, `getItemName, getHeight, getWidth, getSun, getShade`, to access the attributes.

Write, using pseudocode or program code, the procedure `findTree`.

[6]

(b) The programmer will use abstraction to produce the program interface to represent the garden.

i. Give **two** different examples of how abstraction has been used to produce the layout of the garden.

1

2

[2]

ii. Explain the need for abstraction in the production of this program.

[3]

iii. The user needs to input data into the program to set up their garden layout.

Identify **three** pieces of data that the user may input into this program.

1

2 ____

____

3 ____

____ [3]

(c) * The programmer is designing the program to make use of caching and re-useable components.

Explain and evaluate the use of caching and re-useable components in the design of the garden program.

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____ [9]

END OF QUESTION PAPER