1(a) A computer program stores data in an array named `words`.

The data in the array needs to be searched for a value that the user inputs.

i. One example of a searching algorithm is a binary search.

Identify the precondition for a binary search.

---

[1]

ii. A second example of a searching algorithm is a linear search.

Describe how a linear search works.

---

[4]

(b) The array `words` is defined as a global variable and contains these values:

"house"	"boat"	"car"	"telephone"	"garden"	"spice"	"elephant"

The pseudocode function `useWords()` here uses the global array `words`. The number of words in the array `words` is passed as a parameter.

```
function useWords(numberOfWords : byVal)
   contents = ""
   for count = 0 to numberOfWords - 1
     contents = contents + words[count] + " "
   next count
   return contents
endfunction
```

i. Identify **two** variables in the function `useWords()`.

1 _____

2 _____

[2]

ii. `numberOfWords` is a parameter passed by value.

Describe the difference between passing a parameter by value and by reference.

_____

_____

_____

_____ **[2]**

iii. Rewrite the function `useWords()` to use a while loop instead of a for loop.

The function header and close have been written for you.

Write your answer using pseudocode or program code.

```
function useWords(numberOfWords : byVal)
```

_____

_____

```
endfunction
```

[4]

(c) Give **one** benefit and **one** drawback of declaring an array as a global variable instead of a local variable.

Benefit

Drawback

[2]

(d) Describe **one** feature of an Integrated Development Environment (IDE) that can be used to help write a program **and one** feature that can be used to help test a program.

Write

Test

[4]

(e) Functions and procedures are reusable components.

Give **two** benefits of writing a program with reusable components.

1 _____

_____

2 _____

_____

[2]

2(a) A computer program is being written to store data about students.

Fig. 2 shows a binary search tree that stores data about students.

Each student is represented by their ID number. The current contents of the binary search tree are:

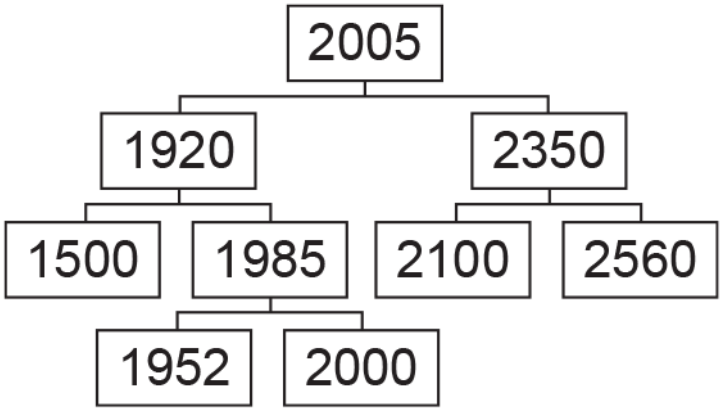


Fig. 2

Identify the root node in the binary tree shown in **Fig. 2**.

_____

[1]

(b) Identify **two** leaf nodes in the binary tree shown in **Fig. 2**.

1 _____

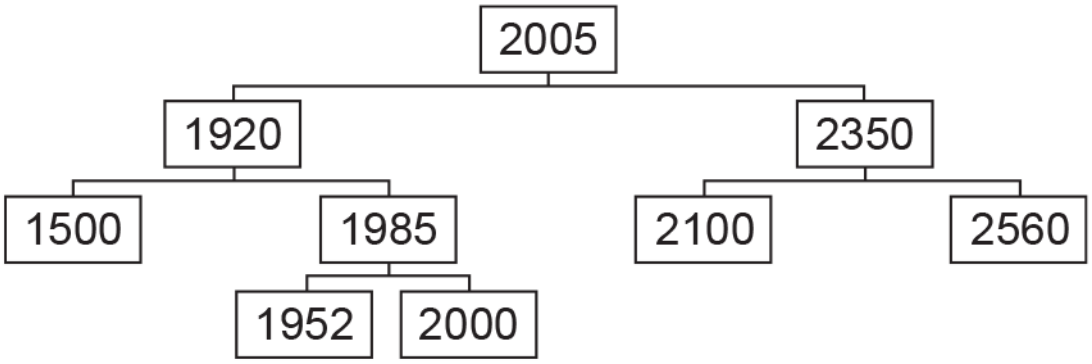
2 _____

[2]

(c) Four more students are added to the binary search tree shown in **Fig. 2** in this order:

1420 2050 2780 2600

Complete the binary search tree here by adding the new students to it.



[4]

(d) * A programmer would like to traverse the binary search tree shown in **Fig. 2**.

Compare the use of a breadth-first traversal and a depth-first (post-order) traversal on the binary search tree.

You should include the following in your answer:

- how each traversal works
- the order of the return values for each traversal.

[9]

3(a) A bubble sort will sort an array of 50 integer values called `numberArray`.

State why the integer values are stored in an array instead of separate variables.

[1]

(b) This bubble sort algorithm is written to sort `numberArray` into ascending numerical order.

Complete this bubble sort algorithm.

```
arrayLength = ..............................................
tempValue = 0
do
   flag = false
   for y = 0 to arrayLength - ..............................................
     if numberArray[y] > numberArray[y + 1] then
      ..................................................... = numberArray[y]
      numberArray[.....................................................] =
numberArray[y + 1]
      numberArray[y + 1] = ..................................................
      flag = true
     endif
   next y
until flag == false
```

[5]

(c) One section of `numberArray` is shown here.

2	12	1	9	3	5	15	7

A second sorting algorithm that could be used to sort this data is a merge sort.

Show how a merge sort will sort this section of the array `numberArray` into ascending numerical order.

[4]

(d) * Another sorting algorithm is insertion sort.

The number of values stored in the array `numberArray` has been reduced to 10.

Compare the use of bubble, merge and insertion sorts on the array `numberArray`.

You should include the following in your answer:

- how each algorithm works
- the Big O complexities for each algorithm
- the suitability of each algorithm for sorting the 10 values.

[12]

4(a) A programmer is developing an aeroplane simulator. The user will sit in a cockpit and the simulated environment will be displayed on screens around them.

The programmer uses computational methods to design a solution for the program.

i. Complete the table by writing a definition for each computational method.

Computational Method	Definition
Abstraction	
Decomposition	

[2]

ii. Give **three** potential differences between the abstracted aeroplane simulator and reality.

1 ----------

---

2 ----------

---

3 ----------

---

[3]

iii. Identify **two** reasons why abstraction is used when designing a solution to the problem.

1 ----------

---

2 ----------

---

[2]

(b) Describe how caching can be used in the aeroplane simulator.

[2]

5(a) **Fig. 5** shows a graph data structure representing a small section of a parcel delivery network. Each node represents an address where deliveries need to be made. The edges show the possible routes and distances between these deliveries.

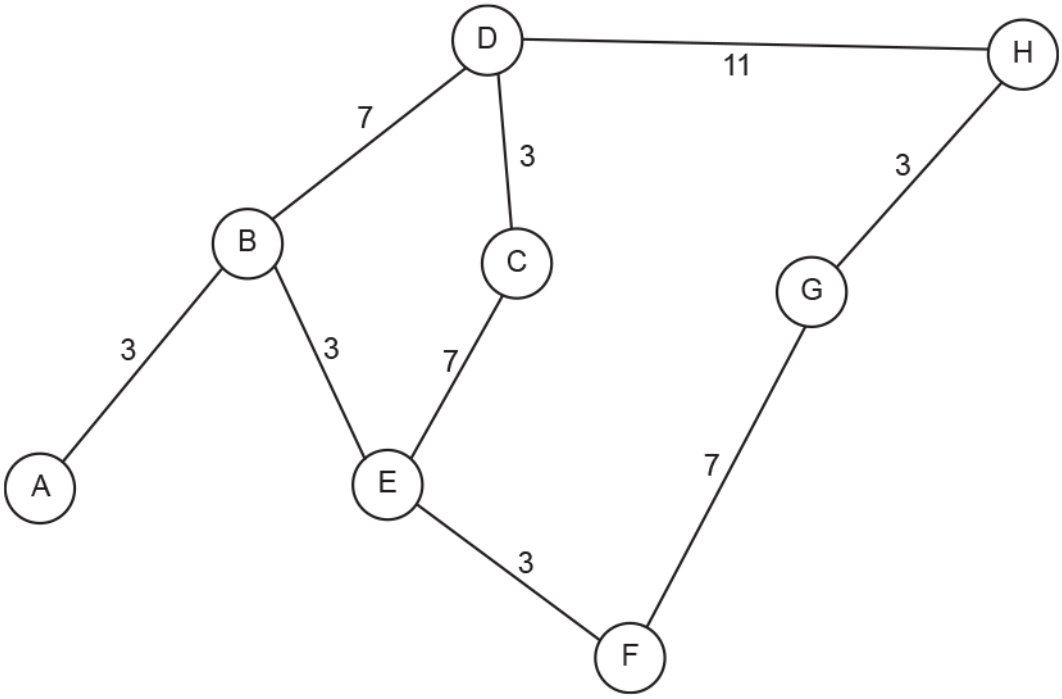


Fig. 5

i. State **one** way a directed graph is different to an undirected graph.

[1]

ii. State **one** way a graph data structure is different to a tree data structure.

[1]

(b) Give **one** reason why the graph is a visualisation of the problem.

---

[1]

(c)

i. Show how Dijkstra's algorithm can be used on the graph shown in **Fig. 5** to find the shortest path from the start node A and the end node H.

You should state the nodes on the final path and the overall distance. Show your working.

You may choose to use the table below to give your answer.

Node	Distance travelled	Previous node

Final path: .....

Distance: ..... [6]

ii. Give a similarity and difference between the performance of Dijkstra's algorithm and the performance of A* algorithm.

Similarity ----------

Difference ----------

---

[2]

(d)

i. State why performance modelling is used to test a system.

---

[1]

ii. Describe how performance modelling can be used in the delivery system.

---

[2]

6(a) A card game uses a set of 52 standard playing cards. There are four suits; hearts, diamonds, clubs and spades. Each suit has a card with a number from; 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

The card game randomly gives 2 players 7 cards each. The unallocated cards become known as the deck.

The players then take it in turns to turn over a card. A valid move is a card of the same suit or the same number as the last card played.

The winner is the first player to play all of their cards.

One component of the game is checking if a move is valid.

Identify **three** other components of the game.

1 ---

2 ---

3 ---

[3]

(b) A function, `checkValid()`, takes the card the player has selected, and the last card played as parameters.

It returns `true` if the player's move is valid and returns `false` if the player's move is not valid.

i. State the reason why `checkValid()` is a function and not a procedure.

---

[1]

ii. The programmer will use a branching (selection) construct to make decisions.

Describe the decisions that will be made in the `checkValid()` function and how these change the return values.

---

[3]

(c) The cards are held in the 2D array `cards`. The first index stores the card number and the second index stores the suit, both as strings.

Write a pseudocode statement or program code to declare the array `cards`.

---

[2]

7(a) A program uses the recursive function `calculate()`. The function is written in pseudocode.

```
1. function calculate(number : byVal)
2.   if number == 1 then
3.     return number
4.   else
5.     return number + calculate (number - 1)
6.   endif
7. endfunction
```

i. Give the line number in the algorithm `calculate()` where a recursive call is made.

---

[1]

ii. State **two** features of any recursive algorithm.

Feature 1 ---

Feature 2 ---

---

[2]

(b) Trace the recursive function `calculate()` and give the final return value, when the following function call is run:

```
calculate(5)
```

You may choose to use the table below to give your answer.

Function call	**number**	**return**
`calculate(5)`		

[5]

(c) Give the pseudocode function call that would return 55 from the recursive function `calculate()`.

---

[1]

8(a) A computer uses a stack data structure, implemented using an array, to store numbers entered by the user.

The array is zero based and has 100 locations.

Fig. 8 shows the current contents of the stack and the first 9 locations of the array.

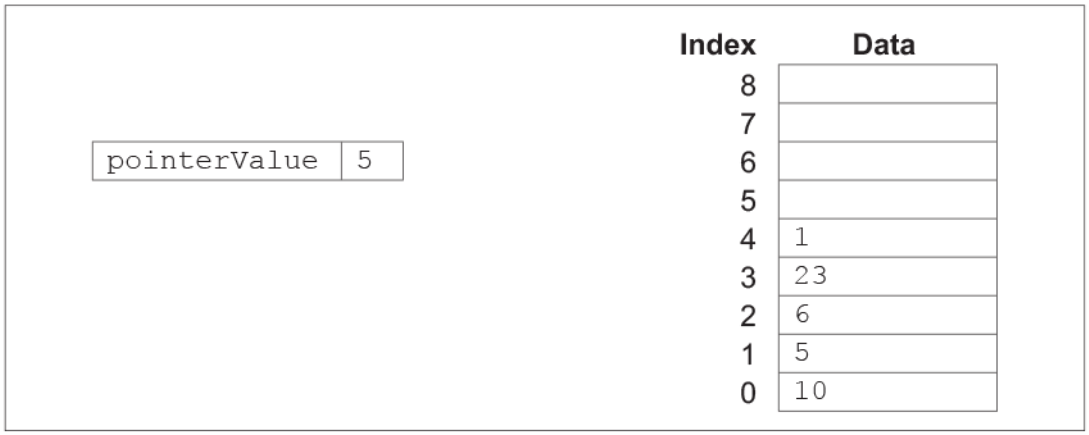


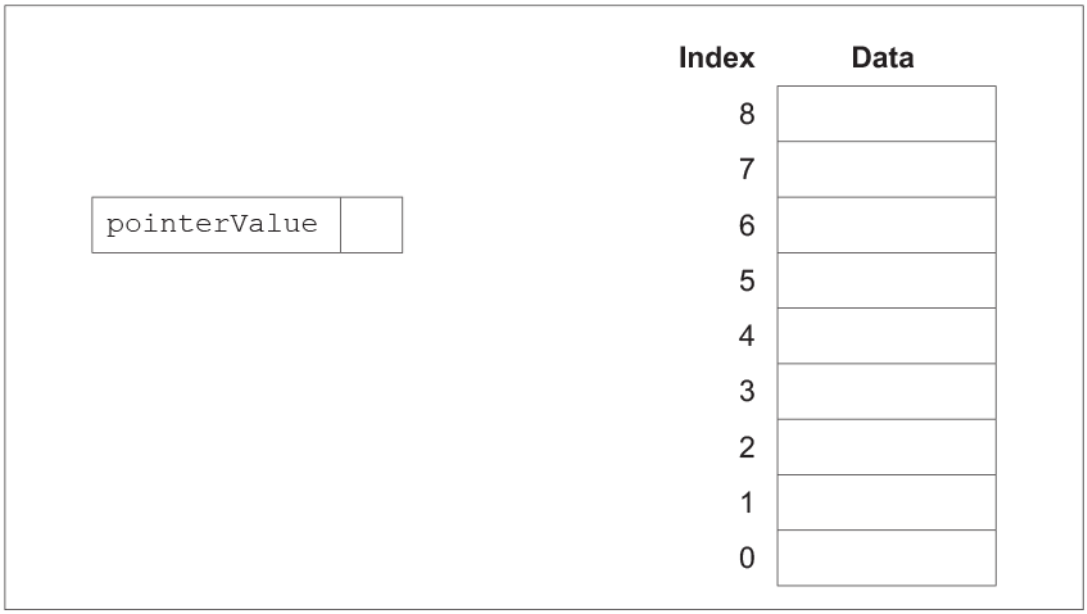
Fig. 8

i. The function `pop()` removes an item from the stack.

The function push() adds an item to the stack that is passed in as a parameter.

Show the contents of the stack and pointer from **Fig. 8** after the following subroutines calls have run.

```
pop()
pop()
push(3)
push(6)
push(7)
```



[2]

ii. State the purpose of `pointerValue`.

---

[1]

(b) The stack is programmed as an object using object-oriented programming. The design for the class, its attributes and methods are shown:

class: stack
attributes: private stackArray : Array of integer private pointerValue : integer
methods: new() function pop() function push(value)

i. The method `pop()` returns the next value in the stack, or `-1` if the stack is empty.

Complete the pseudocode method `pop()`.

```
public function pop()
   if pointerValue == ..... then
     return .....
   else
     pointerValue = pointerValue .....
     returnValue = stackArray[.....]
     return .....
   endif
endfunction
```

[5]

ii. The method push() accepts an integer as a parameter and adds it to the top of the stack unless the stack is already full.

If the push is successful the method returns true.

If the push is unsuccessful due to the stack being full the method returns false.

Write the method push() using either pseudocode or program code.

---

[6]

iii. The main program initialises a new object of type stack with the identifier mathsStack.

Write pseudocode or program code to declare the object.

[2]

iv. The main program needs to:

- take numbers as input from the user
- push them onto the stack `mathsStack` until the stack is full
- output an appropriate message if the stack is full.

Complete the pseudocode algorithm to meet these requirements.

```
returnValue = true
while returnValue == ..............................................
    returnValue = mathsStack.
.......................................(input("Enter Number"))
    if returnValue == ................................................ then
     ................................................ ("Stack full")
    endif
endwhile
```

[4]

v. The main program also needs to:

- remove one item from the stack at a time and add this to a total
- output the total every time an item is removed
- stop removing items when either the stack is empty, or 20 items have been removed.

Write pseudocode or program code to meet these requirements.

[8]

(c) A program is amended to include the use of several queue data structures.

i. Describe how an array can be used to implement a queue data structure.

[3]

ii. * Discuss the use of object-oriented programming and procedural programming to create and manipulate the queue data structures.

You should include the following in your answer:

- the features of object-oriented programming
- the features of procedural programming
- the benefits of using object-oriented instead of procedural programming when creating several queue structures.

[9]

END OF QUESTION PAPER