

You can assume the function `input` takes in a value as an integer.

```
print (answer)
```

[illegible]

. A program stores data in a linked list.

The current contents of the linked list are shown in **Fig. 3**, along with the linked list pointers.

headPointer	1
freeListPointer	4

location	data	pointer
0	"blue"	6
1	"red"	0
2	"green"	8
3	"orange"	NULL
4		5
5		7
6	"grey"	2
7		9
8	"purple"	3
9		NULL

Fig. 3

The function `findNode` will search the linked list and return either the position of the node that contains the data item, or -1 if the data item is not found.

The data held in a node at location `x` can be accessed with `linkedList[x].data`. The pointer of the node at location `x` can be accessed with `linkedList[x].pointer`.

For example, using the linked list shown in **Fig. 3**:

`linkedList[2].data` returns green.

`linkedList[2].pointer` returns 8.

Complete the function, using pseudocode or program code.

```
function findNode(toFind, headPointer, linkedList)

    currentNode = .....

    while(currentNode != ..... )

        if linkedList[currentNode]. ..... == toFind then

            return currentNode

        else

            currentNode = linkedList[.....].pointer

        endif

    endwhile

    return .....

endfunction
```

The function is shown here.

State the value returned from the function when `doCheck(3178)` is called.

[1]

- allow the user to enter an integer value
- pass the value entered into the `doCheck()` function as a parameter
- store both the value input and the value returned from the function in a text file with name "storedvalues.txt"

[5]

4. Sundip writes an algorithm to carry out addition and subtraction. The algorithm will use an initially empty stack with the identifier `numbers` and will take input from the user.

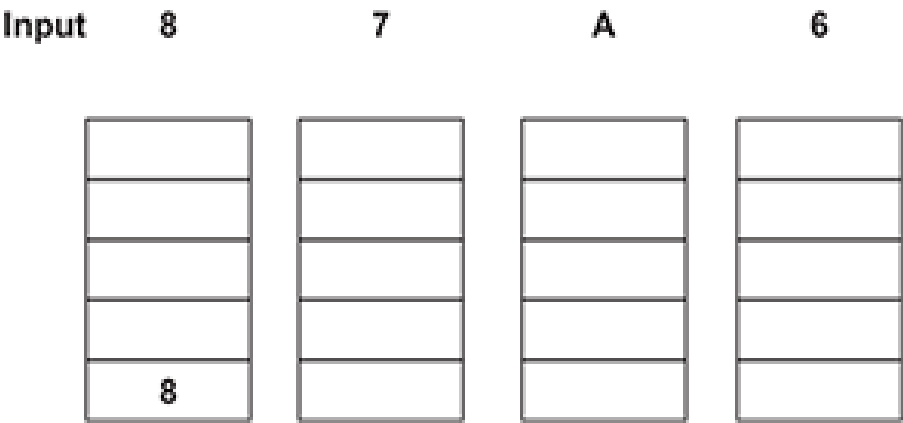
The action the algorithm takes depends on the value input by the user. These actions are listed in **Fig. 2**.

Value input	Action to take
A	• Pop two values from the <code>numbers</code> stack • Add the two values • Push the result back onto the <code>numbers</code> stack
S	• Pop two values from the <code>numbers</code> stack • Subtract the first popped value from the second • Push the result back onto the <code>numbers</code> stack
E	• Pop one value from the <code>numbers</code> stack • Output this value • End program
Any other value	• Push the input value to the <code>numbers</code> stack

Fig. 2

- i. Complete the diagram to show the state of the stack after each value is entered into the algorithm. The letters will complete an action stated in **Fig. 2**.

The state of the stack after the first value, 8, has been completed for you.



ii. Complete the following table to give the output from this algorithm when the following set of inputs are entered by the user. The letters will complete an action stated in **Fig. 2**.

Input data (from left to right)	Output
9 3 A E	
10 5 A 8 S E	
25 5 S 2 3 A S E	

[3]

iii. If the user enters **4 2 S A E** , the algorithm will not work correctly.

Explain what problem this input data will cause and why the problem occurs.

[3]

5. **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.

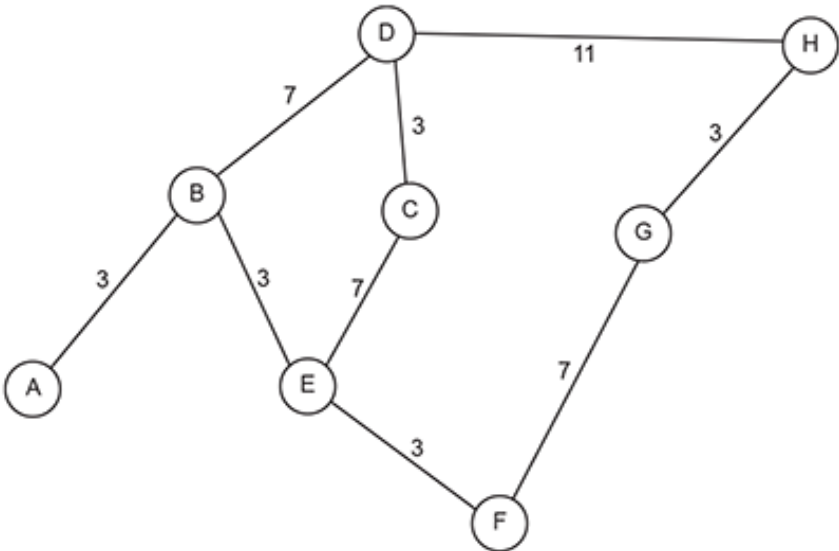


Fig. 5

- i. State 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 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.

You should include the following in your answer:

- how both methodologies could be used to develop a complex computer program
- the benefits of each methodology for this task
- the drawbacks of each methodology for this task.

(b). A programmer creates this function shown in **Fig. 5** using a high-level language.

```
function mystery(x,y)
    total = x + y
    while x >= 10 then
        x = x - 10
        y = y - 10
        total = total + x + y
    endwhile
    return total
endfunction
```

Fig. 5

i. State the value output by the line `print(mystery(10,20))`

[1]

ii. State the value output by the line `print(mystery(0,70))`

[1]

iii. State the value output by the line `print(mystery(45,55))`

[1]

8(a). George owns a small book shop. He wants a program to work out the daily sales figures.

He uses a text file called “Sales.txt” shown in **Fig. 1**. Each line represents the sales total for a different day of the week. The program can run at any point during the week and therefore the text file may not have seven lines.

2367.34
1986.92
2251.49
1882.40
2412.83
3411.32
2721.76

The program needs to read the text file and then calculate:

- The number of days that the program is calculating over
- The total sales over that period
- The average daily sales over that period

At the end of the text file, it should then print the results of these calculations to the screen.

George will use iteration to read through the values in the text file.

Describe how George can use iteration when reading from the text file.

You should write your procedure using pseudocode or program code.

[illegible]

[7]

In order for them to be processed they will need to be cast (i.e. converted) to a different data type.

Explain what data type the values in “Sales.txt” should be converted to.

[2]

An object-oriented system is implemented to organize a company's information about staff attendance. Classes, objects, methods and attributes are used in this system.

Object	
Method	
Attribute	

- ii. Each worker has a name and an attendance figure which can be between 0 and 100.

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are approximately 20 lines visible. The paper has a slightly textured appearance and is set against a dark background.

[5]

	INP	
	STA	numone
	INP	
	STA	numtwo
main	LDA	numone
	SUB	numtwo
	BRP	pos
notpos	LDA	count
	OUT	
	LDA	numone
	OUT	
	HLT	
pos	STA	numone
	LDA	count
	ADD	one
	STA	count
	BRA	main
numone	DAT	
numtwo	DAT	
one	DAT	1
count	DAT	0

Write an algorithm using pseudocode that has the same functionality as the code in **Fig. 1**.

[illegible]

11(a). A website charges a booking fee of £2.99 on each ticket sold. In addition, if the tickets are purchased from outside of the UK, £4.99 is added to the booking fee. The booking fee is calculated using a JavaScript function named `bookingfee()`.

Complete the definition of the `bookingfee()` function below.

```
function bookingfee(numtickets, country) {  
    var nonUKprice = 4.99;  
    var perTicketPrice = .....;  
    var total = 0;  
    if (country!="UK") {  
        total = total + .....;  
    }  
    total = total + (..... * perTicketPrice);  
    ..... total;  
}
```

[4]

(b). The JavaScript function above is used to show users the booking fee. When users click to buy the tickets, the booking fee is calculated again on the server.

- i. Explain why server side processing is used to recalculate the booking fee.

[3]

- ii. Explain **one** advantage of client side processing to either the customer buying the tickets, or to company who own the website.

[2]

```
01 function thisFunction(theArray, num1, num2, num3)
02     result = num1 + ((num2 - num1) DIV 2)
03     if num2 < num1 then
04         return -1
05     else
06         if theArray[result] < num3 then
07             return thisFunction(theArray, result + 1, num2, num3)
08         elseif theArray[result] > num3 then
09             return thisFunction(theArray, num1, result - 1, num3)
10         else
11             return result
12         endif
13     endif
14 endfunction
```

theArray has the following data:

Index:	0	1	2	3	4	5	6	7
Data:	5	10	15	20	25	30	35	40

```
thisFunction(theArray, 0, 7, 35)
```

[illegible]

Function call	num1	num2	num3	result
<code>thisFunction(theArray, 0, 7, 35)</code>				

Final return value

[5]

(b). State the name of the standard algorithm `thisFunction()` performs.

[1]

(c). Hugh could have written `thisFunction()` using iteration instead of recursion.

Compare **two** differences between recursion and iteration.

1

2

[4]

(d). The recursive function `thisFunction()` is printed again here for your reference.

```

01 function thisFunction(theArray, num1, num2, num3)
02   result = num1 + ((num2 - num1) DIV 2)
03   if num2 < num1 then
04     return -1
05   else
06     if theArray[result] < num3 then
07       return thisFunction(theArray, result + 1, num2, num3)
08     elseif theArray[result] > num3 then
09       return thisFunction(theArray, num1, result - 1, num3)
10     else
11       return result
12   endif

```

```
13 endif
```

```
14 endfunction
```

Rewrite the function `thisFunction()` so that it uses iteration instead of recursion.

You should write your answer using pseudocode or program code.

[illegible]

[6]

13. The Little Man Computer (LMC) instruction set can be used to write programs using assembly language.

Fig. 3 shows assembly code written using the LMC instruction set.

	INP	
	STA X	
	SUB Y	
	BRP jump	
	LDA X	
	STA Z	
	HLT	
jump	LDA Y	
	STA Z	
	HLT	
X	DAT	0
Y	DAT	5
Z	DAT	0

Fig. 3

- i. Tick **one** box to indicate the programming construct that is **not** used in **Fig. 3**.

Sequence

☐

Selection

☐

Iteration

☐

[1]

- ii. When the program is run, 7 is input by the user.

State the value that will be in the memory location Z when the program has run with this input.

[1]

- iii. Write an equivalent version of the LMC assembly code shown in **Fig. 3** for a procedural programming language.

You should write your code using pseudocode or program code.

[4]

They will need to record the time that a car enters and leaves their car park in order to calculate the parking charge.

In your answer you should include:

-
- This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and extend across the width of the page. There are no margins, text, or other markings on the paper.