1. A programmer is designing a program that will store data.

The programmer is deciding whether to store the data in a stack or a queue.

The pseudocode function, `enqueue`, inserts an item into a queue.

```
01 function enqueue(item)
02   if tailPointer >= queue.length then
03    return false
04   else
05    queue[tailPointer] = item
06    tailPointer = tailPointer + 1
07    return true
08   endif
09 endfunction
```

i. Give the name of the parameter in the function `enqueue`.

---

[1]

ii. Give the name of **one** global variable that is used in the function `enqueue`.

---

[1]

iii. Describe **one** benefit and **one** drawback of using global variables instead of parameter passing in a subroutine.

Benefit _____

_____

Drawback _____

---

[4]

iv. The function `enqueue` can be called by the main program.

Explain why the function `enqueue` returns true or false values, and how this can be used by the main program that calls the function.

_____

_____

_____

_____

_____

---

[3]

v. The pseudocode function, `dequeue`, removes and returns the first item in the queue. If the queue is empty, the function returns the string `"EMPTY"`.

```
01 function dequeue(data)
02   if headPointer != tailPointer then
03    return "EMPTY"
04   elseif
05    value = queue[headPointer]
06    return value
07    headPointer = headPointer + 1
08   endif
09 endfunction
```

The function dequeue has **several** errors.

Identify the line number of any **three** errors **and** state the correction required.

Error 1 Line Number

Error 1 Correction

Error 2 Line Number

Error 2 Correction

Error 3 Line Number

Error 3 Correction

---

[3]

vi. The programmer has corrected all of the errors in the function `dequeue`.

The main program repeatedly calls the function `dequeue` until all of the elements in the queue have been output.

Write the main program using pseudocode or program code.

---

[3]

2(a). A game is being written that makes use of object-oriented programming. A prototype for one part of the game is being designed that includes a character, a road and a prize to collect.

The road will have 50 spaces that a character can move along. Each space on the road will store a null value or a prize object for the user to collect. Each space is numbered sequentially from the first space (position 0) to the last space (position 49) and will not change during the game. As the player travels down the road, the position the player is on the road will be output.

The road is designed to be a 1-dimensional array with the identifier `road`.

Explain why an array is a suitable data structure to represent the road.

[3]

(b). The characters and prizes are designed as separate classes. 10 of the spaces on the road will contain an instance of the class `Prize`. The other spaces will be empty.

The class design for `Prize` is here.

class: Prize
attributes: private name : string private type : string private value : integer
methods: new() getName() getType() getValue()

`new()` is the constructor method. The name, type and value are passed to the constructor as parameters which then assigns these to the attributes.

i. The method `getName()` returns the data in the attribute `name`.
Write the method `getName()` using pseudocode or program code.

[2]

ii. A global 1-dimensional array, `allPrizes`, stores 10 objects of type `Prize`.

The prize in index 3 has the name “Box”, the type is “money” and the value is 25.

Write pseudocode or program code to create a new object for this prize and store it in index 3 of `allPrizes`.

[3]

iii. The game starts with 10 prizes. Each prize is allocated to one space on the road.

An algorithm needs designing that will generate a random space on the road for each prize. Each road space can only store one prize.

Describe the decisions that will need to be made in this algorithm and how these will affect the program flow.

[3]

(c). The class design for `Character` is here.

class: Character
attributes: private name : string private money : integer private experience : integer private roadPosition : integer
methods: new() getName() getMoney() getExperience() getRoadPosition() changePosition() updateValues()

The four get methods return the associated attribute.

The number of moves is passed to `changePosition()` as a parameter. The method adds this value to the character's position on the road.

The type and value of an object are passed to `updateValues()` as parameters. If the object is money the value is added to the character's money. If the type is experience the value is added to experience. If the type is neither money or experience no changes are made.

i. `new()` is the constructor method. The name of the character is passed into the constructor as a parameter. The constructor then initialises both the experience and road position of the character to 0 and initialises the amount of money to 5.

Write the constructor method for `Character` using either pseudocode or program code.

You do not need to declare the class, the attributes or any other methods.

[5]

ii. The type and value of a prize are passed as parameters to the method `updateValues`. If the type is money the value is added to the character's money. If the type is experience then the value is added to the experience. If the type is neither money or experience no changes are made.

For example, for the Character `player1`:

`player1.updateValues("money",10)` updates player1's money by 10

`player1.updateValues("experience",5)` updates player1's experience by 5

`player1.updateValues("foo",9)` has no effect on player1.

Write pseudocode or program code for the method `updateValues()`.

[5]

(d). This incomplete pseudocode algorithm:

- creates a new character with the name Jamal
- loops until the character reaches the end of the road
- generates a random number of spaces to move between 1 and 4 (including 1 and 4)
- moves the character and checks if the new space has a prize
- updates the character attributes if there is a prize
- outputs the character's new attribute values.

Complete the pseudocode algorithm.

```
character1 = new ..................................... ("Jamal")
newPosition = 0
while newPosition < ...................................
  move = random(1, 4) /this will generate a random number between 1 and 4
  character1.changePosition(move)
  newPosition = character1.getRoadPosition()
  if newPosition < 50 and road[.................................] != null then
    prizeType = road[newPosition].getType()
    valueAmount = road[newPosition].getValue()
    character1.updateValues(.........................., valueAmount)
    print("Congratulations you are in position", newPosition, "and found",
      road[newPosition].getName())
    print("Money =", character1.getMoney(), "and experience =",
      character1. ..................... ())
  endif
.......................
print("You reached the end of the road")
```

[6]

(e). The procedure `displayRoad()` outputs the contents of each space in the road. The number of each space is output with either:

- the word “empty” if there is no prize
- the name of the prize if there is a prize.

```
01  procedure displayRoad()
02     for x = 0 to 60
03       print("Space", y)
04       if road[x] == null then
05        print("empty")
06       elseif
07        print(road[x].getValue())
08       endif
09     next x
10  endprocedure
```

The algorithm contains errors.

Give the line number of **four** different errors and write the corrected line for each error.

Error 1

Error line 1 _____

Correction _____

Error 2

Error line 2 _____

Correction _____

Error 3

Error line 3 _____

Correction _____

Error 4

Error line 4 _____

Correction _____

[4]

(f). A programmer is going to create a prototype for one small part of the game. Both `road` and `allPrizes` will be needed throughout the whole prototype. The programmer is considering making these global arrays as she thinks it will reduce the development time. Another programmer has suggested that doing this may create some problems when the rest of the game is created at a later stage.

Compare the use of global and local variables in this program.

You should include the following in your answer:

- the use of local and global variables
- alternative methods to using global variables
- the appropriateness of each to this program design.

[9]

3(a). A student has written this pseudocode algorithm:

```
01 a = 12
02 do
03   b = input("Enter a number")
04 until b >= 0 and b <= 100
05 for c = 1 to a
06   print(c * a)
07 next c
```

Rewrite lines `05` to `07` to use a while loop instead of a for loop.

You should write your answer using either program code or pseudocode.

[4]

(b). The program uses variables.

i. Describe what is meant by a variable.

[2]

ii. Give the identifiers of all the variables used in this program.

[1]

(c). The student has used a do loop on line `02`.

Describe the difference between a do loop and a while loop.

[2]

4(a). A text-based computer game allows a user to dig for treasure on an island. The island is designed as a grid with 10 rows and 20 columns to store the treasure. Each square is given an x and y coordinate. Some of the squares in the grid store the name of a treasure object. Each treasure object has a value, e.g. 100 and a level, e.g. "Bronze."

A class, `Board`, is used to store the 10 row (x coordinate) by 20 column (y coordinate) grid.

The design for the `Board` class, its attributes and methods is shown here.

class: Board
attributes: private grid : Array of Treasure
methods: new() function getGridItem(x, y) function setGridItem(x, y, treasureToInsert)

The constructor initialises each space in the grid to a treasure object with `value` as -1 and `level` as an empty string.

Complete the following pseudocode for the constructor method.

```
public procedure new()
  for row = ............................. to 9
    for column = 0 to ..............................
      ............................. [row, column] = new
Treasure(.............................,"")
    next ..............................
  next row
endprocedure
```

[5]

(b). A procedure, `guessGrid()`:

- takes a `Board` object as a parameter
- accepts the row (x) and column (y) coordinates from the user
- outputs "No treasure" if there is no treasure found at the coordinate (level is an empty string)
- if there is treasure at that coordinate, it outputs the level and the value of the treasure in an appropriate message.

Write the procedure `guessGrid()` using either pseudocode or program code.

[7]

(c). The main program initialises a new instance of `Board`. The programmer is considering declaring this as a global variable or as a local variable and then passing this into the subroutines that control the game.

Compare the use of variables and parameters in this game.

You should include the following in your answer:

- what is meant by a local variable and global variable
- how local and global variables can be used in this program
- the use of passing parameters by value and by reference.

[9]

5. A programmer has designed a program that includes a reusable program component.

The reusable program component is a function called `isInteger()`. This will take a string as an argument and then check that each digit is between 0 and 9. For example if 103 is input, it will check that the digits 1, 0 and 3 are each between 0 and 9.

The `asc()` function returns the ASCII value of each digit. For example `asc("1")` returns 49.

The ASCII value for 0 is 48. The ASCII value for 9 is 57.

```
01    function isInteger(number)
02       result = true
03       for count = 0 to number.length-1
04          asciiValue = asc(number.substring(count, 1))
05             if not(asciiValue >= 48 and asciiValue <= 57) then
06             result = false
07          endif
08       next count
09       return result
10    endfunction
```

i. Identify **one** identifier used in the function `isInteger()`.

---

[1]

ii. Give the line number where the branching (selection) construct starts in the function `isInteger()`.

---

[1]

iii. Give the line number where the iteration construct starts in the function `isInteger()`.

---

[1]

6. A recursive pseudocode function, `recursiveAlgorithm()`, is shown.

```
01   function recursiveAlgorithm(value)
02      if value <= 0 then
03         return 1
04      elseif value MOD 2 = 0 then
05         return value + recursiveAlgorithm(value - 3)
06      else
07         return value + recursiveAlgorithm(value - 1)
08      endif
09   endfunction
```

Describe the key features of a recursive algorithm.

You may refer to the function, `recursiveAlgorithm()` in your answer.

[3]

7. A programmer uses an Integrated Development Environment (IDE).

Complete the table by identifying **and** describing **three** IDE features that can help the programmer to develop, or debug a program.

IDE feature	Description

[6]

8. Layla writes a pseudocode algorithm to:

- input 20 positive numbers into a 0-indexed 1-dimensional array
- output the average (mean) number as a decimal
- output the smallest number
- output the largest number.

The pseudocode algorithm is shown. It contains various errors.

```
01 total = 1
02 smallest = 9999
03 largest = -1
04 for x = 0 to 21
05   dataArray[x] = input("Enter a number")
06   total = total + dataArray[x]
07   if dataArray[x] < largest then
08      largest = dataArray[x]
09   endif
10   if dataArray[x] < smallest then
11      smallest = dataArray[x]
12   endif
13 next x
14 print("Average = " + total * 20)
15 print("Smallest = " + smallest)
16 print("Largest = " + largest)
```

`dataArray` is defined as a local variable within the main program.

i. State what is meant by a 'local variable'.

_____

_____ **[1]**

ii. Give **one** benefit and **one** drawback of declaring `dataArray` as a local variable in the main program.

Benefit _____

_____

Drawback _____

_____ **[2]**

9. A programmer has designed a program that includes a reusable program component.

Describe the purpose of the following lines in the function `isInteger()`.

Line 03 _____

_____

Line 04 _____

_____

Line 09 _____

[3]

10. Nina is writing a computer game using an Integrated Development Environment (IDE). Her friend, James, is writing a computer game using a text-editor which will allow James to create and edit text. James will use a separate compiler.

Discuss the differences between writing and debugging a program using an IDE and a text-editor.

You should include the following in your answer:

- features that are used when writing code
- features that are used when debugging code
- the benefits of using an IDE instead of a text-editor.

[9]

11(a). A function, `toBinary()`, is needed to calculate the binary value of a denary integer between 0 and 255.

`toBinary()` needs to:

- take an integer value as a parameter
- divide the number by 2 repeatedly, storing a 1 if it has a remainder and a 0 if it doesn't
- combine the remainder values (first to last running right to left) to create the binary number
- return the binary number.

For example, to convert 25 to a binary number the steps are as follows:

25 / 2 = 12	remainder 1
12 / 2 = 6	remainder 0
6 / 2 = 3	remainder 0
3 / 2 = 1	remainder 1
1 / 2 = 0	remainder 1

return value = 11001

Write the function `toBinary()`.

You should write your function using pseudocode or program code.

[6]

(b). The main program:

- asks the user to enter a denary number between 1 and 255
- checks that the input is valid between 1 and 255
- If valid call the function `toBinary()` and pass the input as a parameter
- outputs the return value
- If not valid, repeatedly asks the user to input a number until the number is valid.

Write the algorithm for the main program.

You should write your algorithm using pseudocode or program code.

[4]

12(a). Layla writes a pseudocode algorithm to:

- input 20 positive numbers into a 0-indexed 1-dimensional array
- output the average (mean) number as a decimal
- output the smallest number
- output the largest number.

The pseudocode algorithm is shown. It contains various errors.

```
01 total = 1
02 smallest = 9999
03 largest = -1
04 for x = 0 to 21
05   dataArray[x] = input("Enter a number")
06   total = total + dataArray[x]
07   if dataArray[x] < largest then
08      largest = dataArray[x]
09   endif
10   if dataArray[x] < smallest then
11      smallest = dataArray[x]
12   endif
13 next x
14 print("Average = " + total * 20)
15 print("Smallest = " + smallest)
16 print("Largest = " + largest)
```

i. Identify the construct used on lines 01 to 03 in the algorithm.

_____

[1]

ii. Identify the construct used on lines 10 to 12 in the algorithm.

_____

[1]

(b). Identify **two** variables used in this algorithm.

1 _____

2 _____

[2]

(c). The algorithm that Layla has written has many errors.

Identify the line number of **four** different errors and write the corrected line of code.

Error 1 line number .....

Error 1 correction _____

Error 2 line number .....

Error 2 correction _____

Error 3 line number .....

Error 3 correction _____

Error 4 line number .....

Error 4 correction _____ **[4]**

13(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.

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]

(b). 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]

14(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]

15(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.

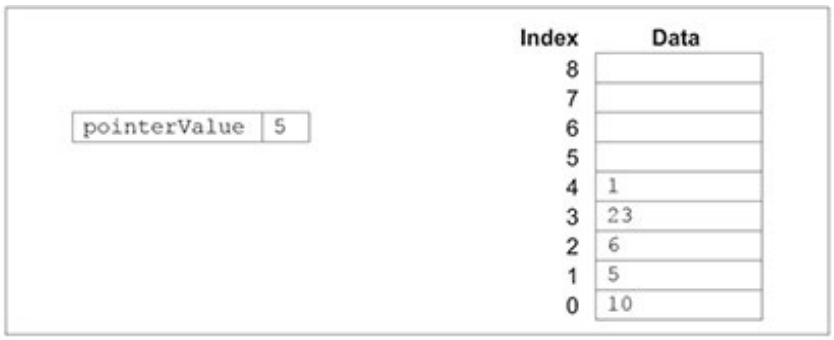


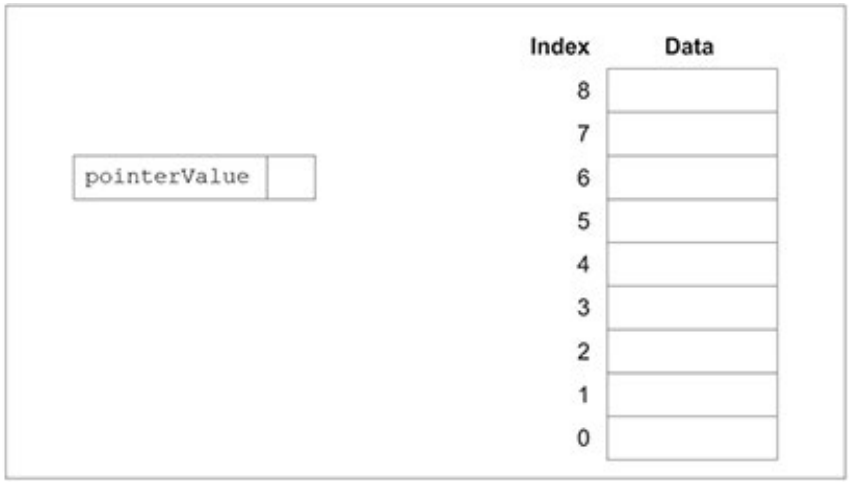
Fig. 8

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]

16. 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.

Describe how caching can be used in the aeroplane simulator.

[2]

17(a). 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]

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

Benefit

Drawback

[2]

(c). 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]

(d). Functions and procedures are reusable components.

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

1

2

[2]

18. Given the following procedure:

```
01 procedure generate(number)
02         a = 0
03         while number > 0
04              if number MOD 2 == 0 then
05                   a = a + 2
06                   print(a)
07                   number = number - 2
08              else
09                   a = a + 1
10                   print(a)
11                   number = number - 1
12              endif
13         endwhile
14 endprocedure
```

State the values printed by the procedure `generate` when `number` = 8.

---

[1]

19. A veterinary surgery uses a two dimensional array to store bookings for customers to bring in their animal to see the vet. There are ten possible booking slots during each day.

An example of the two dimensional array is shown in **Fig. 1**.

- The first column stores the booking slot number, ranging between 1 and 10.
- The second column stores the time of the appointment.
- The third column stores the customerID of the customer who has booked that slot.

1	9:00	5877RC
2	9:30	9655AS
3	10:00	
4	10:30	8754TT
5	11:00	
6	11:30	8745SD
7	13:00	9635GH
8	13:30	
9	14:00	9874PL
10	14:30	9658SV

Fig. 1

If a customerID has been entered for a booking slot then the booking slot has been taken. If no customerID has been entered then the booking slot is available for booking.

When an available time slot has been found then a valid customerID must be entered to confirm the booking.

This is checked by another function called `checkCustomerID`. This will return `true` if the customerID is valid or `false` if the customerID is not valid.

State why a function would be used instead of a procedure for this purpose.

---

[1]

20(a). Kylie buys used games consoles and then sells them to make a profit. She sells her products in multiples of £5 such as £30, £55 and £95. Kylie only accepts £50, £20, £10 and £5 notes from her customers.

Kylie has written an algorithm which will calculate the amount of change needed by stating how many £20, £10 and £5 notes are needed.

The program should output the minimum number of notes required. For example if £35 change is required then it should output 1 x £20 and 1 x £10 and 1 x £5.

```
01 total = input("Enter total price of goods")
02 paid = input("Enter amount paid")
03 global change = paid - total
04 calculateChange()
05
06 procedure calculateChange()
07       twenty = 0
08       ten = 0
09       five = 0
10       while change >= 20 /Calculates number of £20 notes needed
11          twenty = twenty + 1
12          change = change - 20
13       endwhile
14       while change >= 10 /Calculates number of £10 notes needed
15          ten = ten + 1
16          change = change - 10
17       endwhile
18       while change >= 5 /Calculates number of £5 notes needed
19          five = five + 1
20          change = change - 5
21       endwhile
22       print("The amount of change you need is £" + str(change))
23       print("Total £20 Notes:" + str(twenty))
24       print("Total £10 Notes:" + str(ten))
25       print("Total £5 Notes:" + str(five))
26 endprocedure
```

Describe how `calculateChange()` on line 04 is used differently to `calculateChange()` on line 06.

---

[2]

(b). When line 22 is run, it will always print:

```
The amount of change you need is £0
```

Explain why this error occurs when line 22 is run.

---

[2]

(c). Explain why Kylie has used `str` on lines 22 to 25 in her algorithm.

---

[3]

21(a). Ruhail will make use of an Integrated Development Environment (IDE).

State the purpose of an IDE.

[1]

(b). State **two** different programming constructs and give an example of how Ruhail could use each construct when creating his program code.

1

2

[4]

(c). Ruhail has been told to make use of reusable components when creating his program code.

Explain **two** benefits of using reusable components when writing program code.

1

2

[4]

22(a). Logan is writing a program for his customers to be able to buy his gym equipment. In the program, once a customer has selected the items they want to buy, a procedure, `checkDetails`, will be called. This procedure will check that the customer has input their telephone number and also check that it is at least 11 characters long.

Logan has written two possible versions of the procedure that could be used to achieve this.

Version One:

```
procedure checkDetails()
       telephoneNo = input("Enter telephone number")
       while (telephoneNo == "") or (telephoneNo.length < 11)
            print("Error, please try again")
            telephoneNo = input("Enter telephone number")
       endwhile
endprocedure
```

Version Two:

```
procedure checkDetails()
       telephoneNo = input("Enter telephone number")
       if (telephoneNo == "") or (telephoneNo.length < 11) then
            print("Error, please try again")
            telephoneNo = input("Enter telephone number")
       endif
endprocedure
```

i. Explain why version one is more effective than version two at making sure that the telephone number entered is at least 11 characters long.

[4]

ii. As well as the procedure checkDetails, Logan would like to use additional procedures to expand his program.

The program will be expanded to:

- allow customers to be able to register an account by setting up a username and password
- allow registered users to be able to log-in with their registration details
- allow customers, once logged in, to be able to add items that are in stock to their online shopping basket.

State **two** other procedures that Logan could write to meet these requirements, and for each one, state a suitable name and purpose.

Procedure 1:

Procedure Name: _____

Purpose: _____

_____

Procedure 2:

Procedure Name: _____

Purpose: _____

_____

[4]

iii. When setting up the additional procedures in his program, Logan will use a mixture of parameter passing by reference and by value.

State the difference between parameter passing by reference and parameter passing by value.

_____

_____

_____

_____

[2]

(b). *Logan will work in a team with five other programmers and together they will create programming code for a program.

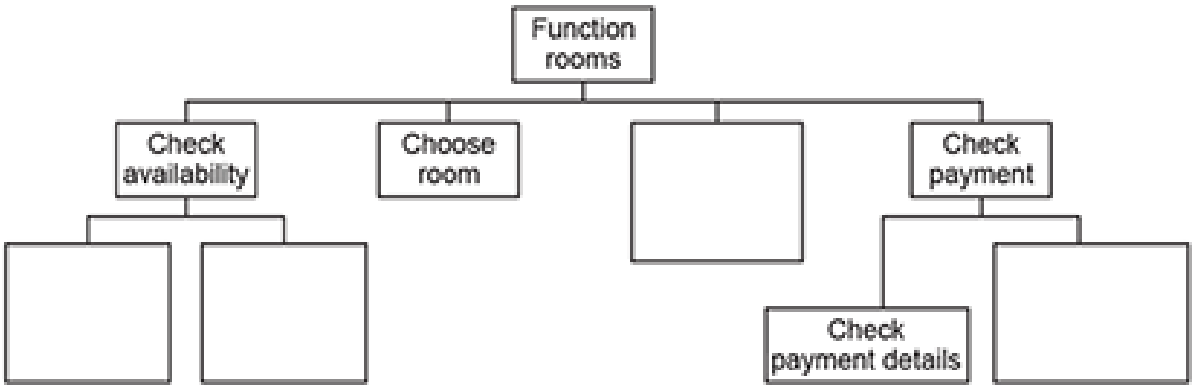
Discuss how modularity can be used to allow the team of programmers to work effectively together on the same program at the same time.

[9]

23. Ruhail owns ten different function rooms which can be hired by different business customers to hold meetings. He would like a program to manage the booking process of each room.

Customers should be able to enter the date they want to hire a function room, and then a list of available rooms will be displayed. Customers can then select which room they want to hire. Customers can then enter their payment details which are then checked and then a confirmation email is sent to the customer.

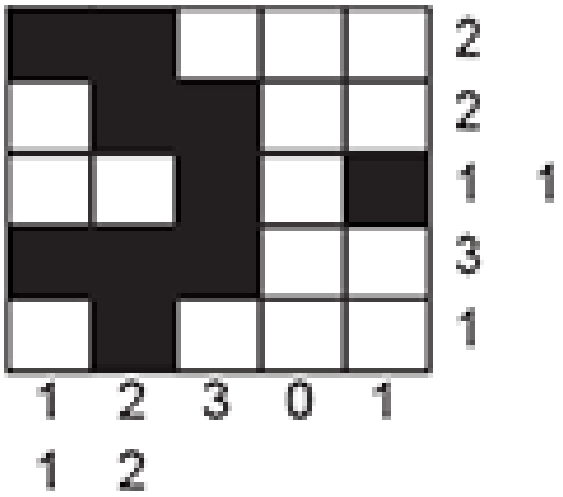
Complete the structure diagram below to show the different component parts of the problem.



[4]

24(a). A Nonogram is a logic puzzle where a player needs to colour in boxes. The puzzle is laid out as a grid and each square needs to be either coloured black or left white.

The numbers at the side of each row and column tells the player how many of the boxes are coloured in consecutively. Where a row has two or more numbers, there must be a white square between the coloured squares.



In this example:

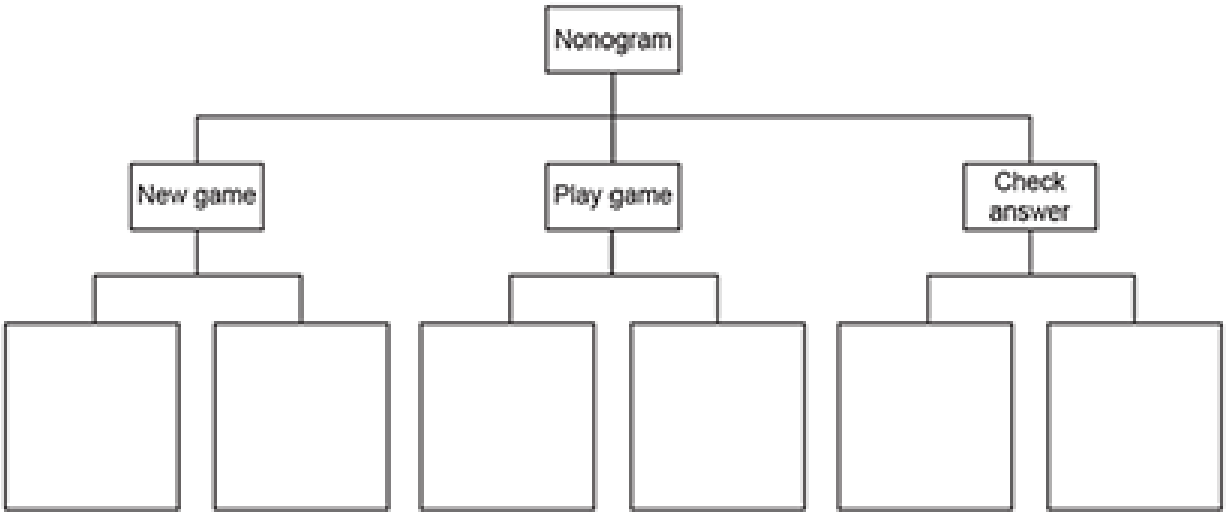
- the first column has 1 1, this means there must be two single coloured boxes in this column. There must be at least 1 white box between them.
- the first row has 2, this means there must be two consecutively coloured boxes in the row.

Juan is creating a program that will store a series of Nonograms for a user to play. The game will randomly select a puzzle and display the blank grid with the numbers for each row and column to the user.

The user plays the game by selecting a box to change its colour. If the box is white it will change to black and if it is black it will change to white. The user can choose to check the answer at any point, and the game will compare the grid to the answers and tell the user if they have got it correct or not.

Juan is creating a structure diagram to design the game.

i. Complete the structure diagram by adding another layer for New game, Play game and Check answer.



[3]

ii. A structure diagram is one method of showing the decomposition of a problem.

Explain why decomposing a problem can help a developer design a solution.

[2]

iii. Identify **one** input, **one** process and **one** output required for the game.

Input

Process

Output

[3]

(b). Juan uses the structure diagram to create a modular program with a number of subroutines.
The program will use two integer 2-dimensional arrays to store the puzzles:

- `puzzle(5,5)` stores the solution
- `answerGrid(5,5)` stores the user's current grid.

A `0` represents a white box and a `1` represents a black box.

i. Juan creates a function, `countRow()`, to count the number of coloured boxes in one row and return the number of consecutive coloured boxes in that row. If there is more than one set of coloured boxes in the row, these are joined together and the string is returned. For example, in the following grid `countRow` for row 0 will return "`2`" as a string, and `countRow` for row 2 will return "`1 1`" as a string. If there are no 1s in a row, then "`0`" is returned as a string.

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Complete the pseudocode algorithm `countRow()`.

```
01    function countRow(puzzle:byref, rowNum:byval)
02    count = 0
03    output = " "
04    for i = 0 to ……
05      if puzzle[rowNum, i] == …… then
06         count = count + 1
07      elseif count >= 1 then
08         output = output + str(……) + " "
09         count = 0
10      endif
11    next i
12    if count>= 1 then
13         output=output+str(count)
14    elseif output == "" then
15         output = "……"
16    endif
17    return ……
18  endfunction
```

[5]

ii. Explain the purpose of line 03 in the function `countRow`.

[2]

iii. Describe the purpose of branching and iteration in the function `countRow`.

[3]

iv. The procedure `displayRowAnswer()` takes `puzzle` as a parameter and outputs the value in each box. Each box in a row is separated by a space. At the end of each row there are two spaces and (by calling the function `countRow` from **part (i)**) the clue values for that row.

For example the puzzle below:

1	1	0	0	0
0	1	1	0	0
0	0	1	0	1
1	1	1	0	0
0	1	0	0	0

Would output:

```
1 1 0 0 0    2
0 1 1 0 0    2
0 0 1 0 1    1 1
1 1 1 0 0    3
0 1 0 0 0    1
```

v. Write pseudocode or program code for the procedure `displayRowAnswer()`.

[6]

vi. The function `checkWon()` takes `answerGrid` and `puzzle` as parameters and compares each element in the grids. If they are identical, it returns `true`, otherwise returns `false`.

```
01  function checkWon(puzzle)
02    for row = 0 to 4
03      for column = 0 to 4
04        if puzzle[row, column] == answerGrid[row, column] then
05          return false
06        endif
07      next column
08    next column
09    return true
10  endfunction
```

There are **three** logic errors in the function `checkWon`

State the line number of each error and give the corrected line.

Error 1 line number _____

Error 1 correction _____

Error 2 line number _____

Error 2 correction _____

Error 3 line number _____

Error 3 correction _____

[3]

(c). * Juan passed the two arrays as parameters, but he did consider making them globally accessible.

Compare the use of global and local variables and data structures in this program. Include the use of parameters and program efficiency in your answer.

[9]

(d). Juan wants to create a program that will generate new Nonograms with different grid sizes. For example a Nonogram with a 10 × 10 grid or a 5 × 20 grid.

Describe how the program could be written to automatically generate a new Nonogram.

[4]

25(a). Lucas writes a program that makes use of a circular queue. The queue stores the data entered into the program. An array is used to represent the queue.

The program needs two pointers to access and manipulate the data in the queue.

State the purpose of the two pointers and give an appropriate identifier for each.

Pointer 1 purpose

Pointer 1 identifier

Pointer 2 purpose

Pointer 2 identifier

---

[4]

(b). Lucas wants a procedure, `enqueue()`, that will add the parameter it is passed to the queue.

Describe the steps the procedure `enqueue()` will follow when adding new items to the queue.

---

[5]

26. * Anna currently writes her program code in a text editor and then runs the compiler.

She has been told that using an Integrated Development Environment (IDE) would be more helpful.

Discuss the benefits of Anna using an IDE to write and test her program rather than using a text editor.

---

[9]

---

27. The pseudocode function `binarySearch()` performs a binary search on the array `dataArray` that is passed as a parameter. The function returns the array index of `searchValue` within the array, and `-1` if it is not in the array.

The pseudocode binary search algorithm is incomplete.

i. Complete the algorithm by filling in the missing statements.

```
function binarySearch(dataArray:byref, upperbound, lowerbound, ……)
  while true
    middle = lowerbound + ((upperbound - lowerbound) ……)
    if upperbound < lowerbound then
      return ……
    else
      if dataArray[middle] < searchValue then
        lowerbound = ……
      elseif dataArray[middle] > searchValue then
        upperbound = ……
      else
        return ……
      endif
    endif
  endwhile
endfunction
```

[6]

ii. The algorithm uses a while loop.

State a different type of loop that could be used instead of the while loop in the given algorithm.

---

[1]

28(a). Hugh has written a recursive function called `thisFunction()` using pseudocode.

```
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
```

The function `DIV` calculates integer division, e.g. 5 DIV 3 = 1

`theArray` has the following data:

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

Trace the algorithm, and give the final return value, when it is called with the following statement:

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

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

Function call	num1	num2	num3	result
`thisFunction(theArray,0,7,35)`				

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.

[6]

29. The following pseudocode procedure performs an insertion sort on the array parameter.

```
01 procedure insertionSort(dataArray:byRef)
02   for i = 1 to dataArray.Length - 1
03     temp = dataArray[i]
04     tempPos = i - 1
05     exit = false
06     while tempPos >= 0 and exit == false
07        if dataArray[tempPos] < temp then
08           dataArray[tempPos + 1] = dataArray[tempPos]
09           tempPos = tempPos - 1
10        else
11           exit = true
12        endif
13     endwhile
14     dataArray[tempPos + 1] = temp
15   next i
16 endprocedure
```

Explain why `dataArray` is passed by reference and not by value.

[2]

30. Bubble sorts make use of two different loops when sorting items into order.

Describe the **two** loops used and their purpose.

1

2

[4]

31(a). Barney is writing a program to store data in a linked list. He is writing the initial program for a maximum of 10 data items.

Each node in the linked list has a data value and a pointer (to the next item).

A null pointer is stored with the value `-1`.

The procedure `printLinkedList()` follows the pointers to print all of the elements in the linked list.

```
01 procedure printLinkedList(headPointer)
02   tempPointer = headPointer - 1
03   dataToPrint = ""
04   if tempPointer == -1 then
05     print("List is full")
06   else
07     while linkedList[pointer].getPointer() != -1
08         dataToPrint = dataToPrint + " " + linkedList[tempPointer,0]
09         linkedList[tempPointer].getPointer() = tempPointer
10     endwhile
11   print(dataToPrint + " " + linkedList[tempPointer].getData())
12   endif
13 endprocedure
```

The procedure has a number of errors.

i. Identify the line of each error and write the corrected line.

Error 1 line number .....

Error 1 correction .....

Error 2 line number .....

Error 2 correction .....

Error 3 line number .....

Error 3 correction .....

[3]

ii. Barney will use an Integrated Development Environment (IDE) to debug his program code.

Describe **three** features commonly found in IDEs that Barney could use to debug his program code.

1

2

3

[6]

(b). * Barney would like his linked list to be part of a base program that is saved in a library. This means that it can be reused and changed by other programs.

Discuss the benefits of using different object-oriented techniques that Barney could use to achieve this.

[12]

32(a). A printer buffer is a storage area that holds the data, known as jobs, that are to be printed by a printer.

A simulation of the printer buffer uses a queue data structure to store jobs that are waiting to be printed. The queue is not circular.

The printer buffer is represented as a zero-indexed 1D array with the identifier `buffer`.

Fig. 2 shows the current contents of the queue `buffer` and its pointers.

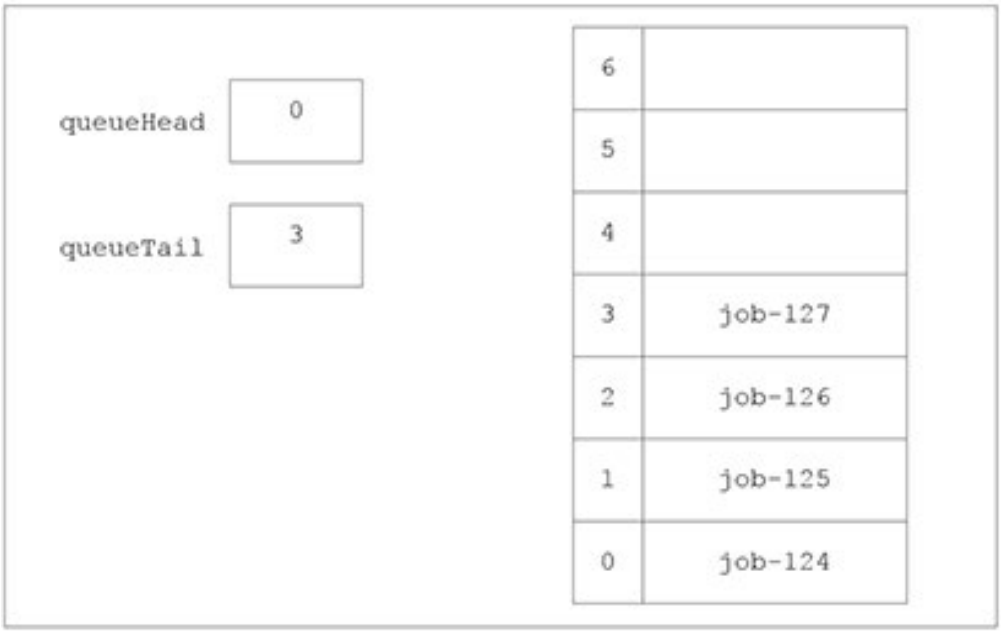


Fig. 2

State the purpose of the pointers `queueHead` and `queueTail`.

`queueHead` _____

_____

`queueTail` _____

_____

[2]

(b). The function `dequeue` outputs and removes the next data item in the queue.

The procedure `enqueue` adds the job passed as a parameter to the queue.

Show the final contents of the queue and pointer values after the following instructions have been run on the queue `buffer` shown in **Fig. 2**.

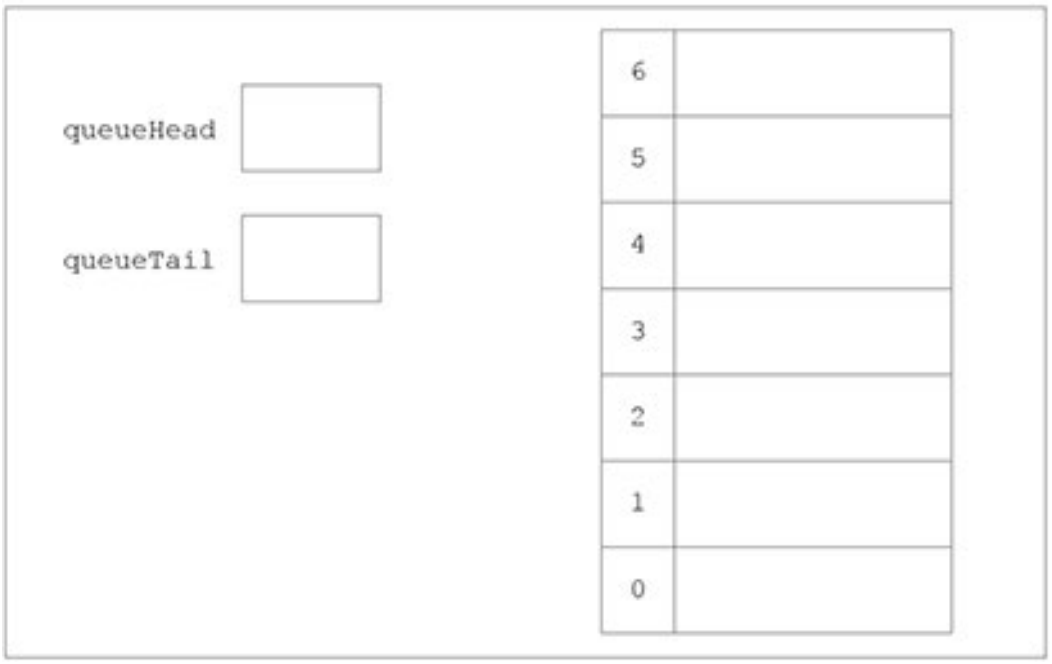
```
dequeue()

dequeue()

enqueue(job-128)

dequeue()

enqueue(job-129)
```



[5]

(c). The array, `buffer` and pointer values are declared with global scope.

i. The function `dequeue` returns `null` if the array is empty, and the contents of the next element if not empty. The queue is not circular.

Write an algorithm, using pseudocode or program code, for the function `dequeue()`.

[5]

ii. The function `enqueue` returns -1 if there is no space at the end of the queue to add data, and returns 1 if the parameter was added to `buffer`. The array `buffer` contains a maximum of 100 elements.

Write an algorithm, using pseudocode or program code, for the function `enqueue()`.

[6]

iii. In the main program of the simulation the user is asked whether they want to add an item to the queue or remove an item.

If they choose to add an item they have to input the job name, and the function `enqueue` is called.

If they choose to remove an item, the function `dequeue` is called and the job name is output.

Appropriate messages are output if either action cannot be run because the queue is either empty or full.

Write, using pseudocode or program code, an algorithm for the main program of the simulation.

[8]

(d). The queue is changed to make it a circular queue.

Describe how the functions `enqueue` and `dequeue` will need to be changed to allow `buffer` to work as a circular queue.

[3]

(e). Some print jobs can have different priorities. The higher the priority the sooner the job needs to be printed.

Describe how the program could be changed to deal with different priorities.

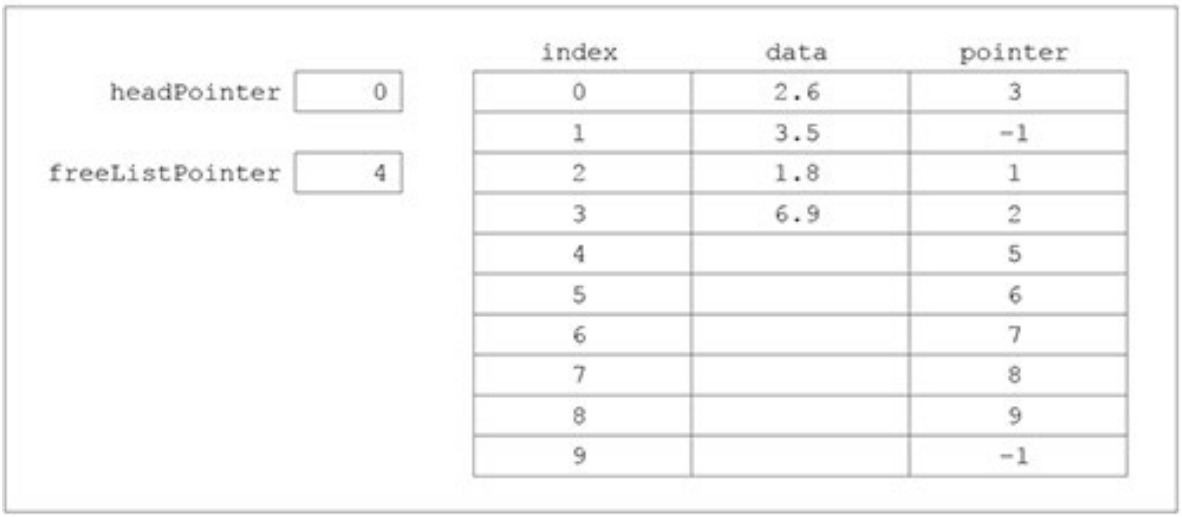
[3]

33(a). Barney is writing a program to store data in a linked list. He is writing the initial program for a maximum of 10 data items.

Each node in the linked list has a data value and a pointer (to the next item).

A null pointer is stored with the value `-1`.

Fig. 3 shows the current contents of the linked list including the head and free list pointer values.



headPointer	0
freeListPointer	4

index	data	pointer
0	2.6	3
1	3.5	-1
2	1.8	1
3	6.9	2
4		5
5		6
6		7
7		8
8		9
9		-1

Fig. 3

i. Describe the purpose of `freeListPointer`.

[2]

ii. State the purpose of `headPointer`.

[1]

iii. Show the contents of the linked list from **Fig. 3** and the pointer values when the node with data `6.9` is deleted.

headPointer

freeListPointer

index	data	pointer
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		

[4]

(b). Barney wants the nodes to be stored as objects using object-oriented programming. He designs the following class.

class: node
attributes: private data : Real private pointer : Integer
methods: new (newData, newPointer) getData() getPointer() setData(newData) setPointer(newPointer)

The constructor assigns the parameters to the attributes to create an object.

i. Write an algorithm, using pseudocode or program code, to create the class `node`, its attributes and constructor.

You do **not** need to write the get and set methods.

[4]

ii. The class `node`, uses get methods and set methods.

Describe **one** difference between get methods and set methods.

[2]

(c). The function `findNodePath()` takes the data item to find in the linked list as a parameter and follows the pointers to find the required node.

The function returns the array indexes of all the nodes it visits and joins this to a suitable message stating whether the data was found or not found and then returns this as one string.

Describe how the function `findNodePath()` will search for the data item and return the required message.

[6]

34. Oscar owns a taxi company. He would like a program to handle taxi bookings from customers.

Some of Oscar's customers are rated as gold. Customers who are rated as gold are given priority when they make a taxi booking. Some customers rated as gold are shown here.

Arshad	**Betty**	**Dave**	**Freddie**	**Harry**	**Jimmy**	**Kanwal**	**Lynn**	**Siad**	**Tommy**	**Will**

When a customer makes a booking, Oscar will make use of a binary search to check if they are gold rated.

Oscar would like to know if 'Tommy' is gold rated.

i. State the **three** values that will be set as the midpoints and then checked against 'Tommy' on each iteration of the binary search.

Show your working here.

Midpoint 1 _____

Midpoint 2 _____

Midpoint 3 _____

[3]

ii. Oscar has 75 000 customers stored in his program.

Describe the benefit to Oscar of using binary searches in his program.

Benefit _____

_____

_____

[2]

iii. State **one** other search algorithm that Oscar could have used.

_____

[1]

iv. State the pre-condition which has been met, which meant that Oscar did not need to use the search algorithm you stated in the part above.

[1]

35. Daisy is a computer technician. She is responsible for making sure all new employees are given a username to access the computer network.

The rules that are followed when creating a new username are as follows:

Step 1: The employee's first name is entered (e.g. Roger)

Step 2: The employee's surname is entered (e.g. Banks)

Step 3: A username is then made up from:

- Their whole surname (e.g. Banks)
- The first letter of their first name (e.g. R)
- A number 1

For example: BanksR1

Step 4: The username is then checked against existing usernames. This is done by calling a function `existingUsers`. This will return `true` if the username is unique and `false` if the username already exists.

Step 5: If the username is unique then "Username is Unique" should be printed. If the username already exists then the number at the end of the username should increase by one (e.g. BanksR2).

Step 6: Steps 4 and 5 should be repeated until the username is unique.

Write a procedure called `createUsername` that meets the rules of Daisy's program.

You should write your procedure using pseudocode or program code.

[9]

36(a). Sally is a classroom teacher. She would like a program to be able to organise where students will sit in her classroom.

A plan of her classroom is shown **in Fig. 1**.

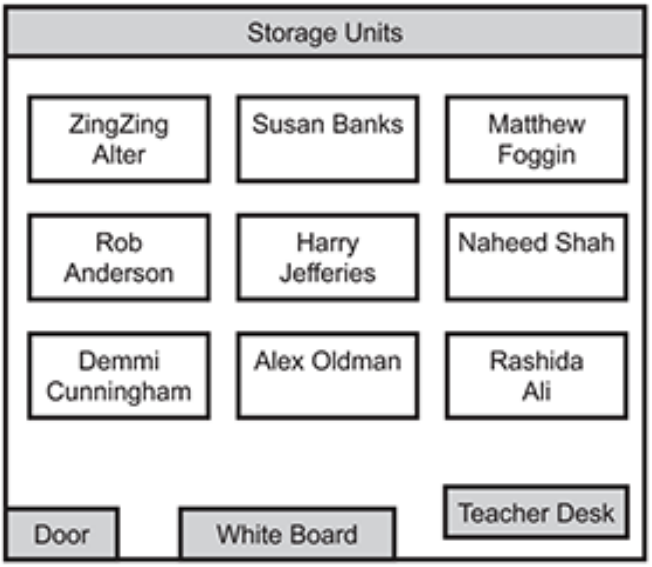


Fig. 1

Sally would like to increase the security of her program by adding a password to enter the program. She has created the procedure, `checkPassword`, to do this.

```
01   procedure checkPassword()
02       correctPassword = "ComputerScience12"
03       check = false
04       while check == false
05          enteredPassword = input("Enter Password")
06          if enteredPassword == correctPassword then
07              check = true
08          endif
09       endwhile
10   endprocedure
```

i. Identify the programming construct used on lines 06 to 08 in the procedure `checkPassword`.

[1]

ii. Sally has used a `while` loop on line 04 of the procedure `checkPassword`.

Explain why Sally has used a `while` loop instead of a `for` loop.

[4]

iii. Sally could have used a `do until` loop instead of a `while` loop.

Rewrite lines 04 to 09 of the procedure `checkPassword` using a `do until` loop instead of a `while` loop.

[3]

(b). Sally will make use of an Integrated Development Environment (IDE) to create her program code.

i. Describe **three** features that are commonly found in IDEs that will help Sally write her program code.

1

2 _____

3 _____

[6]

ii. Sally uses a Rapid Application Development (RAD) approach when creating her program.

Describe **two** benefits of using RAD.

1 _____

2 _____

[4]

iii. Sally will make use of an appropriate test strategy to test her programming code.

Compare **one** difference between black box testing and white box testing.

_____

[2]

END OF QUESTION PAPER