1. Eve enjoys playing board games. Her favourite board game is called “Pot Luck”. This has a numbered grid of 10 squares by 10 squares. Each square has a number between 1 and 100.

Players place their game counters on square 1. A 30-minute timer is set which counts downwards. Each player rolls two 6-sided dice and then moves their game counter that number of squares. Some squares tell the player to pick up a card. These have instructions on, such as 'Move forward 10 spaces'. If the player lands on one of these squares they move according to the instruction on the card. The first player to land on square 100, is announced as the winner. If no winner is announced before the timer runs out, then it is a draw.

Logical conditions are checked once a player has rolled the dice.

Describe **two** different logical conditions and how the result will affect the outcome of the game.

Logical Condition 1

Condition: ______________________________

____

Outcome: ______________________________

____

Logical Condition 2

Condition: ______________________________

____

Outcome: ______________________________

______________________________ **[4]**

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 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]

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

Line 03

Line 04

Line 09

[3]

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

```
function recursiveAlgorithm(value)
  if value <= 0 then
    return 1
  elseif value MOD 2 = 0 then
    return value + recursiveAlgorithm(value - 3)
  else
    return value + recursiveAlgorithm(value - 1)
  endif
endfunction
```

Trace the recursive function, `recursiveAlgorithm()`, and give the final return value when called with `recursiveAlgorithm(10)`. You may choose to use the table below to give your answer.

Function call	`value`	`return`

Final return value ....

[5]

5.

i. The array `numbers` contains 356 numbers to be sorted by the bubble sort algorithm.

State the maximum number of passes a bubble sort would need to complete to sort 356 numbers into order.

[1]

ii. State the name of **one** other sorting algorithm.

[1]

6. Taylor is designing a program for a client who would like to simulate earthquakes on major cities around the world in 3D. The client would like to be able to view any stage of an earthquake such as:

1. the build-up of the earthquake
2. the earthquake taking place
3. the aftershocks of the earthquake.

The client would also like to be able to play the simulation at different speeds. For example, a slow, normal or fast speed.

The program will need to accept inputs from the user before playing the simulation.

i. Identify **two** different inputs for this program.

1 --

----

2 --

-- **[2]**

ii. One decision point in the program will be to decide if the user inputs are suitable or not.

Identify **two** other example decision points in this program.

1 --

----

2 --

-- **[2]**

7. A program uses a bubble sort to sort data into ascending numerical order.

The data is stored in a 0-indexed 1-dimensional array.

Show each stage of a bubble sort to sort this data into ascending numerical order:

1	5	3	9	2	7

You should clearly show and label each pass in your answer.

[4]

8(a). 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
```

Explain why = is used on line 11 of the procedure `generate` instead of ==.

[2]

(b). State the values printed by the procedure `generate` when `number` = 7.

[1]

(c). Describe the purpose of the `MOD` operator on line `04` of the procedure `generate`.

[2]

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

```
procedure insertionSort(dataArray:byRef)
  for i = 1 to dataArray.Length - 1
    temp = dataArray[i]
    tempPos = i – 1
    exit = false
    while tempPos >= 0 and exit == false
       if dataArray[tempPos] < temp then
          dataArray[tempPos + 1] = dataArray[tempPos]
          tempPos = tempPos - 1
       else
          exit = true
       endif
    endwhile
    dataArray[tempPos + 1] = temp
  next i
endprocedure
```

State whether the procedure `insertionSort` sorts the data into ascending or descending order and explain your choice.

[3]

10(a). Poppy would like to use a bubble sort to sort 250 000 numbers into order from lowest to highest.

Currently the first five numbers before they have been sorted are:

195 584	167 147	158 187	160 125	184 236

Currently the last five numbers before they have been sorted are:

1058	19 558	1915	20 215	15

* Discuss how a bubble sort works and how efficient it will be when sorting these 250 000 items into order from lowest to highest.

[9]

(b). State the number of comparisons that will need to be made in the first pass of the bubble sort.

[1]

11(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.

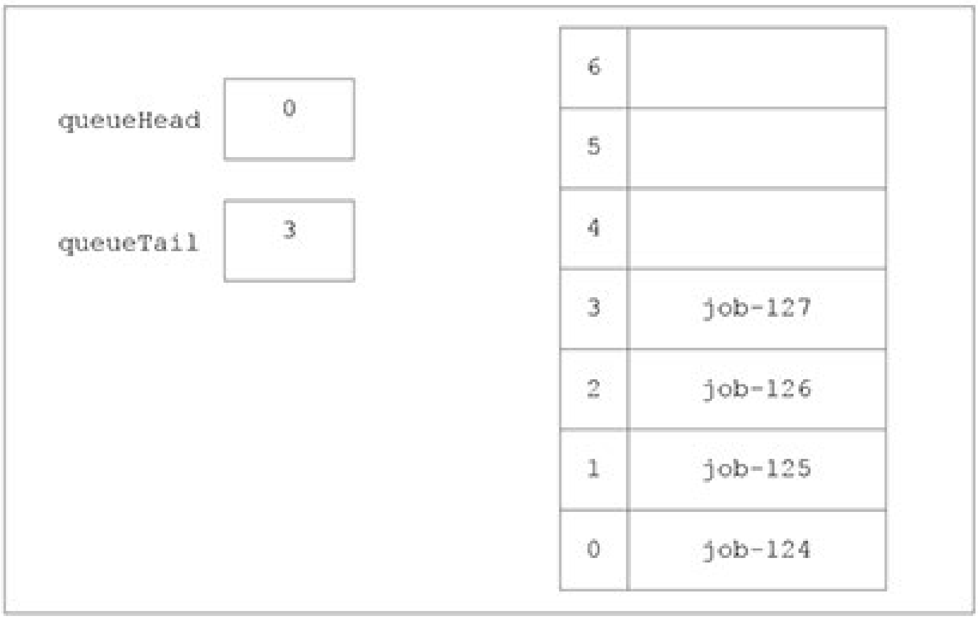

Fig. 2

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)
```

queueHead	
queueTail	

6	
5	
4	
3	
2	
1	
0	

[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]

12(a). Given the following procedure:

```
procedure maths(number)
    a = (number DIV 10) * 10
    b = a + 10
    if (number - a) >= (b - number) then
       print(b)
    else
       print(a)
    endif
endprocedure
```

State the value printed by the procedure `maths` if `number=10` ______________ **[1]**

(b).

State the value printed by the procedure `maths` if `number=27` ______________ **[1]**

(c).

State the value printed by the procedure `maths` if `number=14` ______________ **[1]**

END OF QUESTION PAPER