1(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
```

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]

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

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

[2]

(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]

2 A company runs a Virtual Learning Environment (VLE). Schools can register students to use the VLE. The students get their own account and the school can view and monitor their students who are registered. There are currently over 10 000 schools registered, each with up to 1000 students.

The students can watch videos, take quizzes and communicate using forums and online chat tools.

The company gathers a large amount of data and wants to use data mining to help them decide how to improve the VLE in the future.

Discuss how the company can use data mining to decide how to improve the VLE.

You should include the following in your answer:

- the characteristics of data mining
- the benefits of data mining in this scenario
- the drawbacks of data mining in this scenario.

[12]

3(a) A computer game has a building containing 7 rooms. There are secret passages between each room. **Fig. 3** shows the rooms and the passages between the rooms represented as a graph data structure.

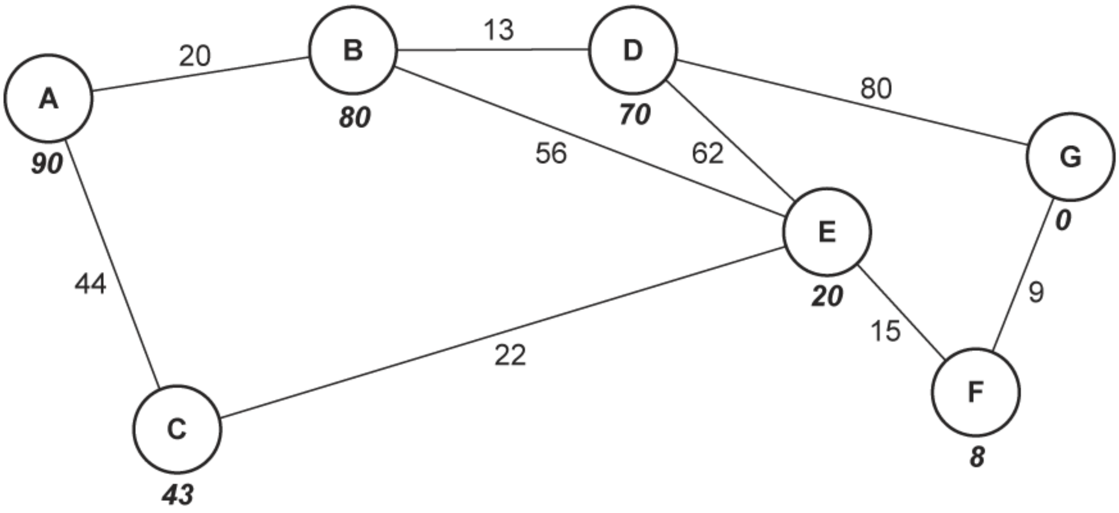


Fig. 3

The number in bold below each node in **Fig. 3** is the heuristic value.

Perform an A* algorithm on the graph shown in **Fig. 3** to find the shortest path from A to G.

Show your working, the nodes visited and the distance.

You may use the table below to give your answer.

Node	Distance travelled	Heuristic	Distance travelled + Heuristic	Previous node

Final path: ....

Distance: ....

[7]

(b) State **four** ways that a graph data structure is different from a tree data structure.

1

2

3

4

[4]

(c) The final game will involve multiple computer-controlled characters and interactive elements that make use of artificial intelligence to determine the moves they will make.

The artificial intelligence will use heuristics to determine where the computer-controlled characters will move in the game.

Discuss how heuristics can be used in algorithms.

You should include the following in your answer:

- the purpose of heuristics
- the benefits and drawbacks of heuristics
- the suitability of heuristics in algorithms within a computer game.

[9]

4(a) The current contents of a queue data structure are shown in **Fig. 4**.

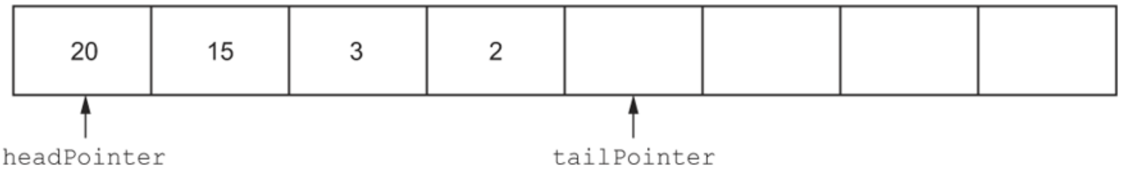


Fig. 4

State the purpose of `headPointer` and `tailPointer` in the queue shown in **Fig. 4**.

`headPointer`

`tailPointer`

[2]

(b) `enqueue` will add data to the queue. `dequeue` will remove data from the queue.

Show the contents of the queue and the position of both pointers after the following actions have been executed on the queue shown in **Fig. 4** in the order given:

- `enqueue(20)`
- `dequeue()`
- `dequeue()`

[2]

(c) The queue is used to store ID numbers of jobs that a program needs to process. Some jobs will be given a priority which means they need to be processed first.

Explain why this queue is **not** a suitable data structure for this program.

[2]

5(a) The contents of a stack are stored in the 1-dimensional array called `numbers`.

`topStack` stores the index of the next free space in the stack.

The array is declared with space for 100 elements.

The function `pop()` returns the next item from the stack and updates the appropriate pointers.

Describe the steps in the function `pop()`.

[4]

(b) The function `push()` inserts its parameter called `dataValue` onto the stack and updates the appropriate pointers.

Complete the function `push()` using pseudocode or program code.

```
function push(... ......................................)
  if .................................. ! = 100 then
  numbers [..................................] = dataValue
  topStack = topStack + ...................................
  return true
  else
  return false
  endif
endfunction
```

[4]

(c) Write an algorithm, using pseudocode or program code, to call the function `push()` with the value 15 and output a message saying “`Added`” if the value was successfully inserted onto the stack or “`Not Added`” if the stack is full.

---

[4]

6(a) The following strings are stored in an array.

“rainbow”	“moon”	“sun”	“stars”	“clouds”	“tornado”

Explain how a linear search would search the array for the index that stores “clouds”.

[3]

(b) State why a binary search cannot be used in this example.

[1]

(c) Show how an insertion sort will sort the given data into **ascending** alphabetical order.

“rainbow”	“moon”	“sun”	“stars”	“clouds”	“tornado”

[5]

(d)

i. A sorting algorithm has a best **time** complexity of O(n).
Describe what is meant by the best **time** complexity O(n) for a sorting algorithm.

[2]

ii. Another sorting algorithm has a worst **space** complexity of O(log(n)).

Describe what is meant by the worst **space** complexity O(log(n)) for a sorting algorithm.

[2]

iii. Identify the **time** complexity that means the time will not change even when the number of items increases.

[1]

iv. Identify the **space** complexity that means the amount of memory (space) used will double each time a new item is included.

[1]

7(a) A computer game stores tasks that the player has requested. Each task has:

- an identification (ID) number e.g. **Task A**
- a real number to be processed e.g. **123456.789**
- an integer number to represent the order the tasks should be accessed e.g. **1**.

The task that needs to be processed the earliest is given the order number 1.

Two or more tasks can have the same order number. For example, two tasks can have an order number 1.

The data about each task needs to be stored. This will store the ID number, data value and order number for a task.

Explain why a record data structure is suitable for this data.

[2]

(b) The tasks will be stored in a binary search tree before they are processed. They are stored in ascending order by their order number.

i. Give **two** characteristics of a binary search tree.

1

2

[2]

ii. Give an advantage of storing the tasks in a binary search tree instead of a 1-dimensional array.

[1]

iii. Tick (✓) **one** column in each row to identify whether each statement applies to a depth-first (post-order) tree traversal, a breadth-first tree traversal, or neither of these two traversals, when performed on a binary search tree.

Statement	Depth-first (post-	Breadth-first	Neither of these two

	order)		traversals
All nodes at the current depth are visited before moving to the next depth			
The algorithm traverses to the end of one branch before moving to another branch			
The algorithm will make use of backtracking			
The traversal can be used to output the contents of the tree in ascending order			
The algorithm will output the root node last			

[5]

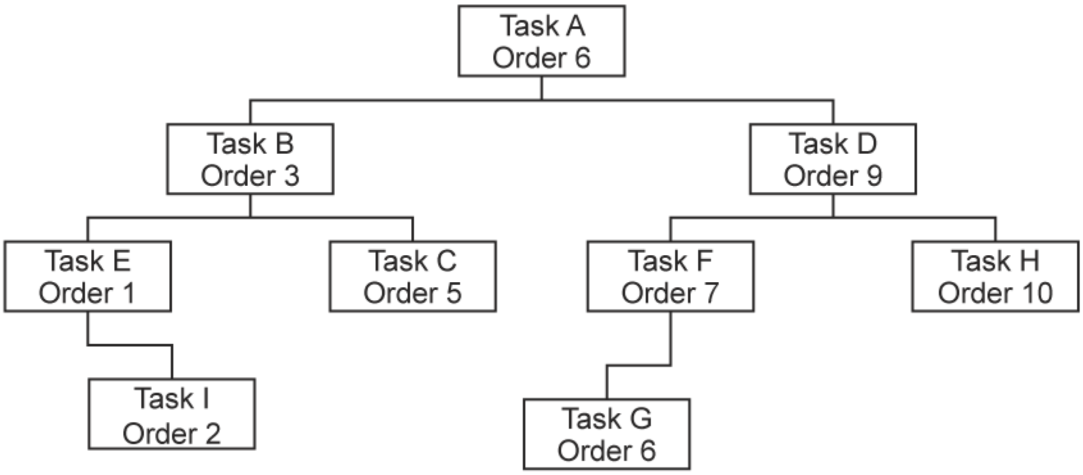
iv. The tasks currently stored in the binary search tree are shown here.

When a new task is inserted with the same order number as a pre-existing task, it is classed as having a higher order number.

For example, task G has the same order number as task A. Since task G was inserted after task A it is classed as a higher number.

Change the diagram to show the contents of the binary search tree after the following tasks are inserted in the order given:

- Task X with order number 12
- Task Y with order number 7
- Task Z with order number 11



[3]

8(a) A group of students are designing a racing car game. The game will allow players to enter their name and then a choice of vehicle. They will then race against other vehicles that will be controlled by the program. Players will use the arrow keys to control their vehicle.

The students are identifying the inputs and outputs for the game.

Complete the table by identifying **two** inputs and **two** outputs for the game.

Input 1	....
Input 2	....
Output 1	....
Output 2	....

[4]

(b) The students use abstraction during the design process.

i. State what is meant by abstraction **and** describe how it can be used to design the racing car game.

Definition ____

____

Use ____

____

____

____ **[3]**

ii. Explain why it is beneficial to use abstraction when designing a computer program such as a game.

____

____

____

____

____

____ **[3]**

(c)

i. The group of students use decomposition.

State what is meant by decomposition.

---

[1]

---

ii. Describe **one** benefit of using decomposition when designing a computer program such as a game.

---

[2]

9(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]

END OF QUESTION PAPER