

**Modified Enlarged 24pt
OXFORD CAMBRIDGE AND RSA EXAMINATIONS**

Tuesday 12 October 2021 – Afternoon

A Level Computer Science

H446/02 Algorithms and programming

**Time allowed: 2 hours 30 minutes
plus your additional time allowance**

YOU CAN USE:

a ruler (cm/mm)

an HB pencil

DO NOT USE:

a calculator

Please write clearly in black ink.

Centre number

Candidate number

First name(s) _____

Last name _____

READ INSTRUCTIONS OVERLEAF



INSTRUCTIONS

Use black ink.

Write your answer to each question in the space provided. If you need extra space use the lined pages at the end of this booklet. The question numbers must be clearly shown.

Answer ALL the questions.

INFORMATION

The total mark for this paper is 140.

The marks for each question are shown in brackets [].

Quality of extended response will be assessed in questions marked with an asterisk (*).

ADVICE

Read each question carefully before you start your answer.

SECTION A

Answer ALL the questions.

- 1 Taylor is creating an online multiplayer game where users can create accounts and build their own circus. Each circus will contain characters such as clowns, animals, magicians and dancers.**

Users can set up a new circus in the online world, purchase new characters and visit other users' circuses.

(a) Taylor uses computational methods to analyse the problem including abstraction.

Describe how Taylor could use abstraction in the design of his online circus game.

[3]

(b) Taylor will make use of concurrent processing within his circus game.

(i) Describe what is meant by the term 'concurrent processing'.

[2]

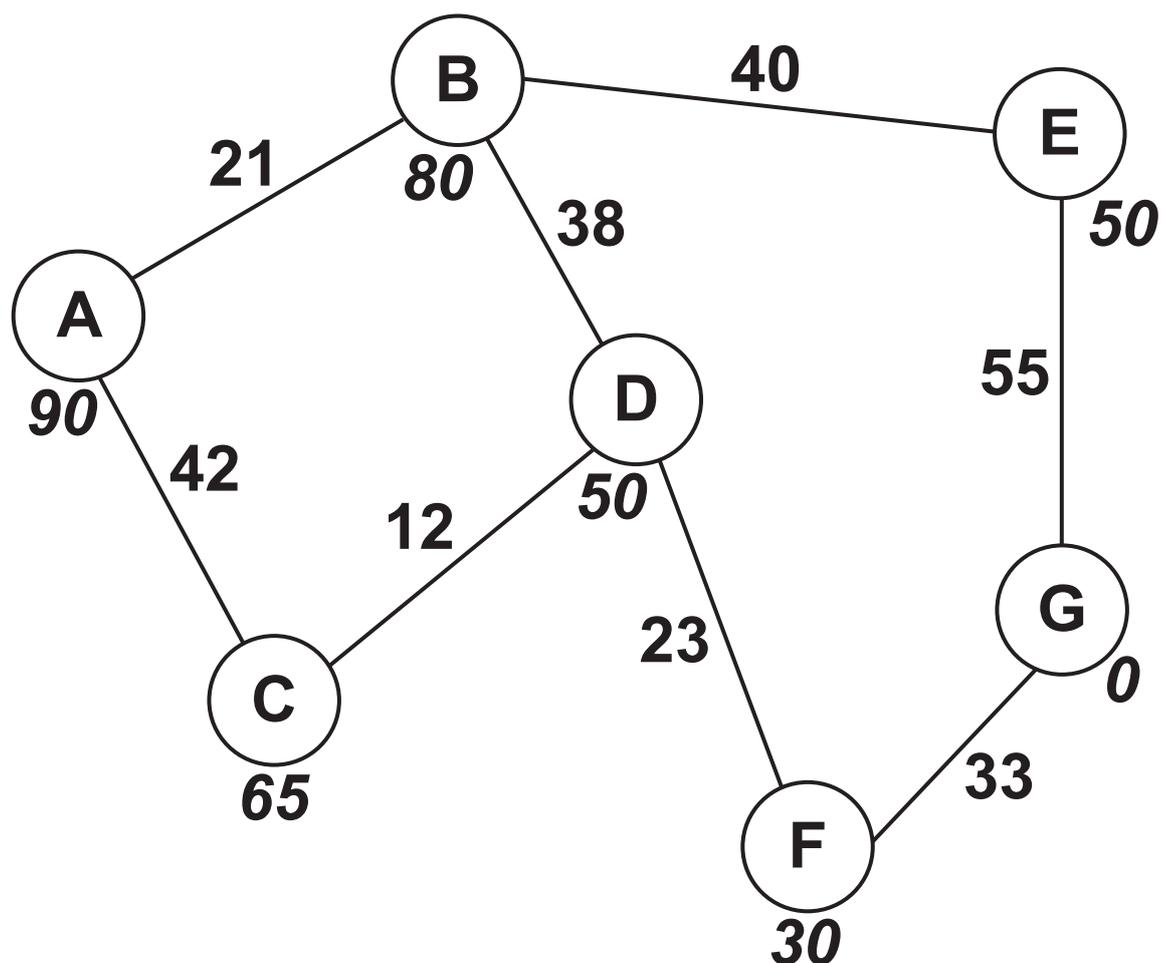
(ii) Explain why concurrent processing is needed to allow multiple users to log in and interact with game elements at the same time.

[3]

(c) Some of the characters in the game will move and interact independently. Taylor is going to use graphs to plan the movements that each character can take within the game.

DancerGold is one character. The graph shown in FIG. 1 shows the possible movements that DancerGold can make.

FIG. 1



DancerGold's starting state is represented by node A. DancerGold can take any of the paths to reach the end state represented by node G.

The number on each path represents the number of seconds each movement takes.

The number in bold below each node is the heuristic value from A.

- (i) Define the term heuristic in relation to the A* algorithm.**

[2]

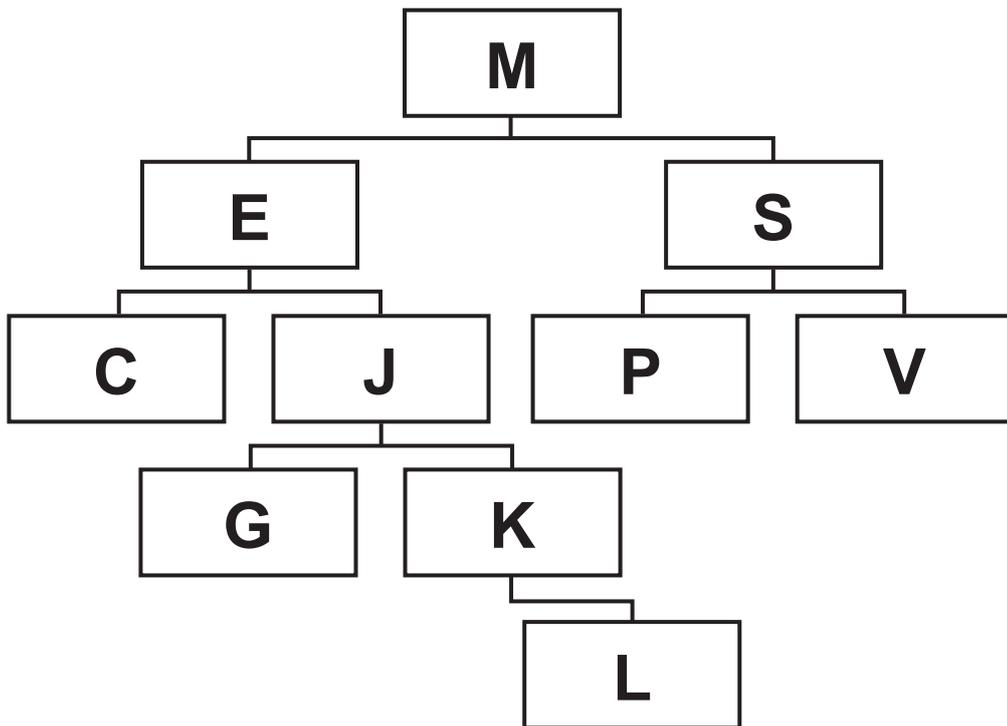
Node	Distance travelled	Heuristic	Distance travelled + Heuristic	Previous node

Final path: _____

Distance: _____

(d) A breadth-first traversal can be performed on both a tree and a graph.

Show how a breadth-first traversal is performed on the following binary tree.



[6]

(e)*The game will have thousands of users. Taylor will store data about the users and their actions while playing the game in a large database.

Evaluate how Taylor can use data mining to inform future changes to improve his circus game. [9]

2 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.

(a) The pseudocode binary search algorithm is incomplete.

(i) Complete the algorithm by filling in the missing statements. [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]

```

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

```

- (b) The tables below show possible Big O complexities for the worst-case space, best-case space and average time for search algorithms.

Tick the worst-case space complexity for a binary and linear search. [6]

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

Tick the best-case space complexity for a binary and linear search.

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

Tick the average time complexity for a binary and linear search.

	Binary search	Linear search
$O(\log(n))$		
$O(1)$		
$O(n)$		

(c) Identify ONE situation where a linear search is more appropriate than a binary search.

[1]

(b) Explain why a quicksort is known as a divide and conquer algorithm.

[2]

4* 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.

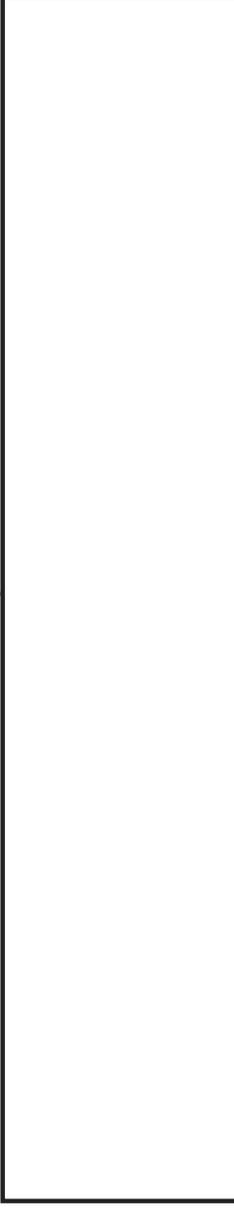
Discuss the benefits of Anna using an IDE to write and test her program rather than using a text editor. [9]

BLANK PAGE

5 Christoff is writing a program to simulate a city using object-oriented programming. He is designing classes to store different types of buildings and their location on the road. He has created the following plan for some of the buildings:

```
class name : building
attribute:
private numberFloors
private width
private height
methods:
new (pFloors, pWidth, pHeight)
function getNumberFloors ()
function getWidth ()
function getHeight ()
function setNumberFloors (pFloors)
function setWidth (pWidth)
function setHeight (pHeight)
```





```

class name : office
attributes:
private numDesks :
integer
private numCompanies :
integer
methods:
new (pFloors, pWidth,
pHeight, pDesks,
pCompanies)
function getDesks ()
function getCompanies ()
function
setDesks (pDesks)
function
setCompanies (pCompanies)

```

```

class name : house
attributes:
private bedrooms
private bathrooms
methods:
new (pFloors, pWidth,
pHeight, pBedrooms,
pBathrooms)
function getBedrooms ()
function getBathrooms ()
function
setBedrooms (pBedrooms)
function
setBathrooms (pBathrooms)

```

(a) The method `new` is used to denote the constructor for each class.

State the purpose of the constructor.

[1]

The classes `office` and `house` inherit from `building`.

(b) Describe what is meant by inheritance with reference to these classes.

[2]

(c) Part of the declaration for the class `building` is shown.

Complete the pseudocode declaration by filling in the missing statements. [5]

```

class building
private numberFloors
private width
private .....
public procedure new(pFloors, pWidth, pHeight)
    numberFloors = .....
    width = pWidth
    height = pHeight
endprocedure
public function getNumberFloors ()
    return .....
endfunction
public function setNumberFloors (pFloors)
    //sets the value of numberFloors when the
    parameter is >= 1

```

```
//returns true if numberFloors is successfully
changed,
//returns false otherwise
if pFloors >= 1 then
    numberFloors = .....
    return true
else
    return .....
endif
endfunction
endclass
```

(e) Christoff develops a new class to store the houses in one road. His class design is shown:

```
class : houseRoad
```

```
attributes:
```

```
private buildings(100) //array of  
class house
```

```
private numberBuildings //records  
the number
```

```
//of houses currently stored in  
the array
```

```
//buildings
```

```
methods:
```

```
new (building)
```

```
function getBuilding (buildingNum)
```

```
procedure newbuilding (pBuilding)
```


(f) Christoff wants to create a new house called houseOne. It has the properties: 2 floors, 8(m) width, 10(m) height, 3 bedrooms and 2 bathrooms.

The house is located on a road with the identifier limeAvenue of type houseRoad, houseOne is the first house in this road.

Write pseudocode or program code to declare the house houseOne, road limeAvenue and assign houseOne to the first array position in the road. [4]

6 Amy's processor makes use of pipelining during the fetch-decode-execute cycle.

(a) The processor's pipeline consists of the following stages:

- Fetching the instruction from memory**
- Decoding the instruction**
- Executing the instruction.**

Instructions A, B, C and D need to be processed.

Identify the stage(s) and instruction(s) run during each pipeline below. [4]

Pipeline 1 _____

Pipeline 2 _____

Pipeline 3 _____

Pipeline 4 _____

(b) Explain why pipelining can improve the performance of the processor.

[2]

7 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.

(a) 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 _____

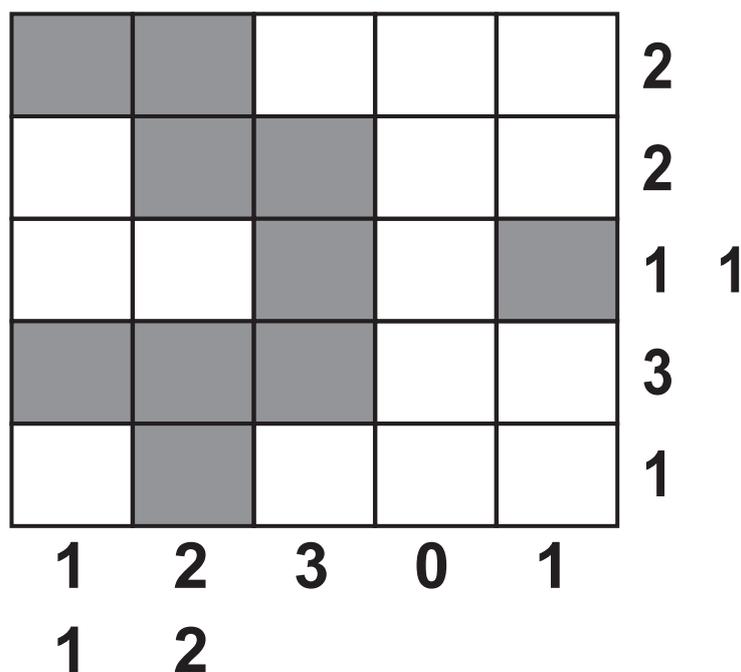
[4]

SECTION B

Answer ALL the questions.

- 8 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 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 grey and if it is grey 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.

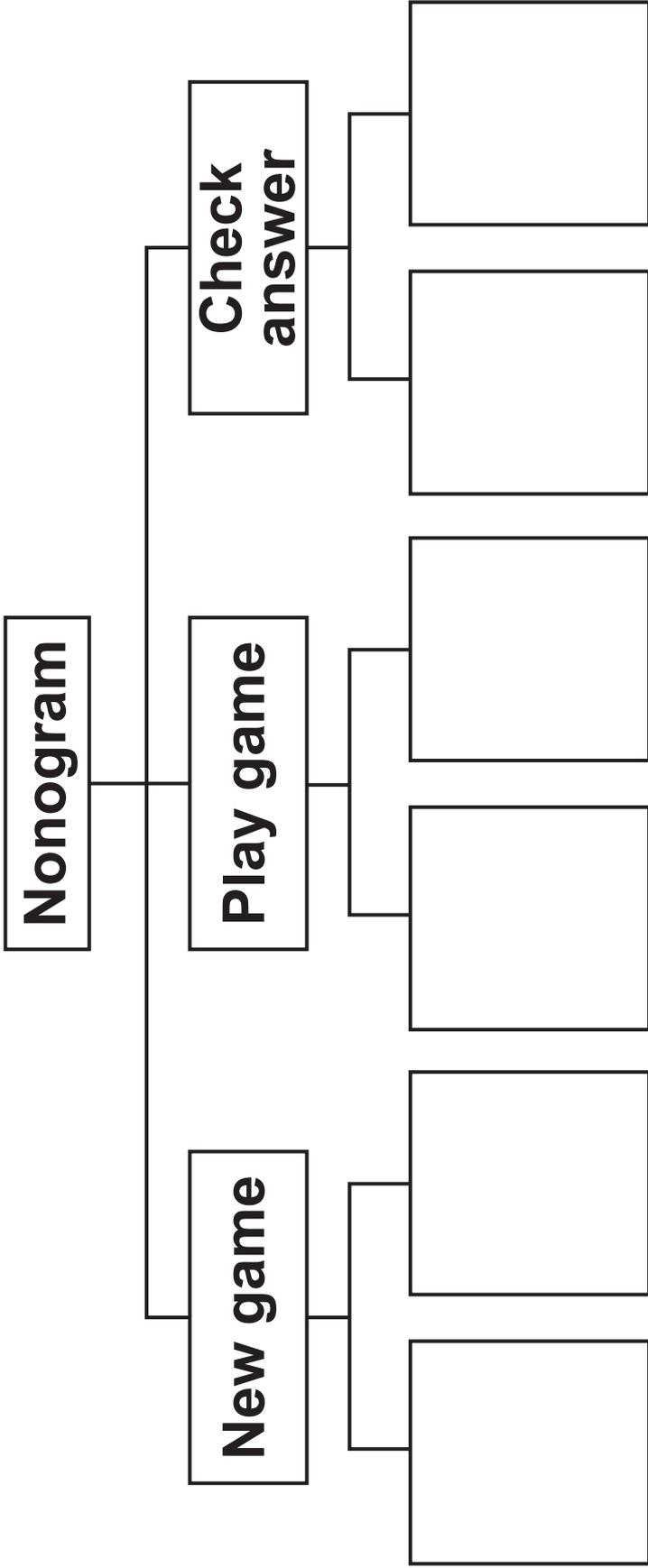
(a) Juan is creating a structure diagram to design the game.

(i) Complete the structure diagram opposite 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 grey 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()` opposite. [5]

```

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

```

(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]

BLANK PAGE

(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 8(b)(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) 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`.

There are **THREE** logic errors in the function `checkWon` opposite.

State the line number of each error and give the corrected line.
[3]

Error 1 line number _____

Error 1 correction _____

Error 2 line number _____

Error 2 correction _____

Error 3 line number _____

Error 3 correction _____

```
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
```


(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]

END OF QUESTION PAPER

