OCR
Oxford Cambridge and RSA

A Level Computer Science

H446/02 Algorithms and programming

Friday 15 June 2018 – Morning

Time allowed: 2 hours 30 minutes

7015026144

You may use:
- a ruler (cm/mm)
- an HB pencil

Do not use:
- a calculator

H446/02

First name									
Last name									
Centre number						Candidate number				

INSTRUCTIONS
- Use black ink.
- Complete the boxes above with your name, centre number and candidate number.
- Answer **all** the questions.
- Write your answer to each question in the space provided. Additional paper may be used if required but you must clearly show your candidate number, centre number and question number(s).
- Do **not** write in the barcodes.

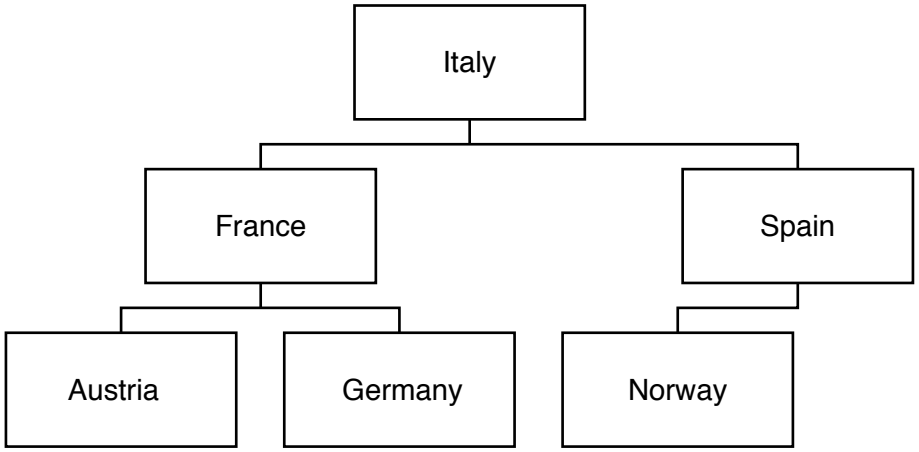
INFORMATION
- The total mark for this paper is **140**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended responses will be assessed in questions marked with an asterisk (*).
- This document consists of **28** pages.

Section A

Answer **all** the questions.

1 A program stores entered data in a binary search tree.

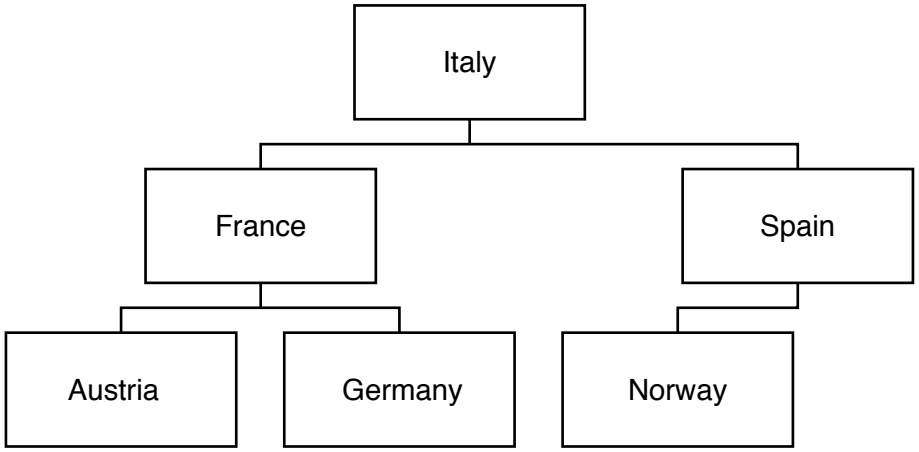
The current contents of the tree are shown:



(a) Complete the diagram to show the contents of the tree after the following data is added:

England, Scotland, Wales, Australia **[3]**

(b) Show the order of the nodes visited in a breadth first traversal on the following tree.



……

……

……

……

……

……

……

…...……**[3]**

(c) A pseudocode algorithm is written to search the tree to determine if the data item "Sweden" is in the tree.

The function `currentNode.left()` returns the node positioned to the left of `currentNode`.

The function `currentNode.right()` returns the node positioned to the right of `currentNode`.

```
function searchForData(currentNode:byVal, searchValue:byVal)
    thisNode = getData(……)
    if thisNode == …… then
        return ……
    elseif thisNode < searchValue then
        if currentNode.left() != null then
            return (searchForData(currentNode.left(), searchValue))
        else
            return ……
        endif
    else
        if ……  != null then
            return (searchForData(currentNode.right(), searchValue))
        else
            return false
        endif
    endif
endfunction
```

(i) Complete the algorithm.

[5]

(ii) The algorithm needs to be used in different scenarios, with a range of different trees.

Identify **two** preconditions needed of a tree for this algorithm to work.

1 .....

2 .....

[2]

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

2 A company merger is joining five e-commerce retailers under one company, OCRRetail. Each retailer has a different sales system and OCRRetail wants to develop one computer system that can be used by all the retailers.

Mary's software development company has been employed to analyse and design a solution for the company.

(a) (i) Two computational methods (techniques used to solve a problem using computational thinking) that Mary will use are problem recognition and decomposition.

State what is meant by problem recognition and decomposition.

Recognition .....

.....

Decomposition .....

..... **[2]**

(ii) State **one** additional computational method.

..... **[1]**

(b) Mary plans to use data mining to generate information about OCRRetail's customers. Mary will use this information to benefit the company.

(i) Define the term 'data mining'.

.....

..... **[1]**

(ii) Identify **two** pieces of information that data mining could provide OCRRetail about sales, and state how OCRRetail could make use of this information.

1 .....

.....

.....

.....

2 .....

.....

.....

..... **[4]**

(c) Mary has developed the program and is considering using performance modelling before installing the system.

(i) Define the term 'performance modelling'.

.....

..... **[1]**

(ii) Identify **one** way performance modelling could be used to test the new system.

.....

..... **[1]**

(d) Mary created the program as a series of sub-programs that can be reused.

Describe **one** benefit of Mary creating reusable program components.

.....

.....

.....

..... **[2]**

3 A puzzle has multiple ways of reaching the end solution. Fig. 3 shows a graph that represents all possible routes to the solution. The starting point of the game is represented by A, the solution is represented by J. The other points in the graph are possible intermediary stages.

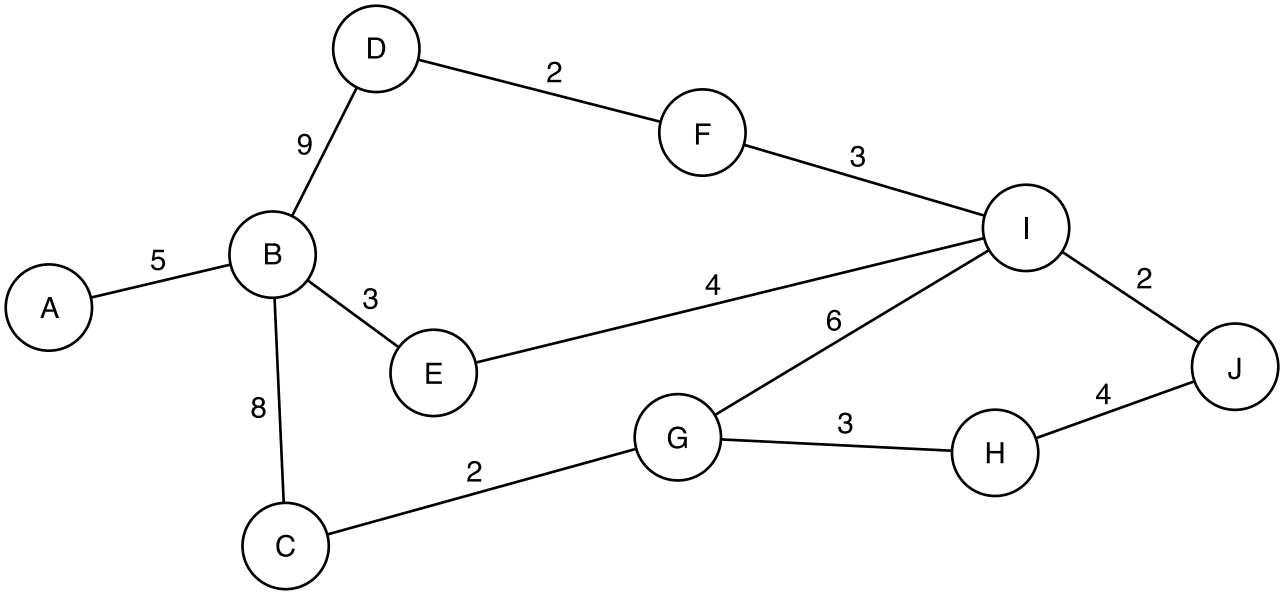


Fig. 3

(a) The graph in Fig. 3 is a visualisation of the problem.

(i) Identify **one** difference between a graph and a tree.

.....

..... **[1]**

(ii) Explain how the graph is an abstraction of the problem.

.....

.....

.....

..... **[2]**

(iii) Identify **two** advantages of using a visualisation such as the one shown in Fig. 3.

1 .....

.....

2 .....

..... **[2]**

(b) Demonstrate how Dijkstra's algorithm would find the shortest path to the solution in Fig. 3.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[7]**

(c)* The creator of the puzzle has been told that the A* algorithm is more efficient at finding the shortest path because it uses heuristics.

Compare the performance of Dijkstra's algorithm and the A* search algorithm, making reference to heuristics, to find the shortest path to the problem.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

(d) A computer program version of the puzzle is to be developed. A programmer will use an IDE to debug the program during development.

Describe **three** features of an IDE that help debug the program.

1.....

.....

.....

.....

2.....

.....

.....

.....

3.....

.....

.....

.....

[6]

4 A recursive function, `generate`, is shown.

```
function generate(num1:byval)
    if num1 > 10 then
        return 10
    else
        return num1 + (generate(num1 + 1) DIV 2)
    endif
endfunction
```

(a) Trace the algorithm to show the value returned when `generate(7)` is called. Show each step of your working.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[6]**

(b) The parameter, `num1`, is passed by value.

Explain why the parameter was passed by value instead of by reference.

.....

.....

.....

..... **[2]**

(c)* Parameters can be used to reduce the use of global variables.

Compare the use of parameters to global variables in recursive functions.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[9]**

(d) A student called Jason writes a recursive algorithm. The recursive algorithm uses more memory than if Jason had written it as an iterative algorithm.

Explain why the recursive algorithm uses more memory than the iterative algorithm.

.....

.....

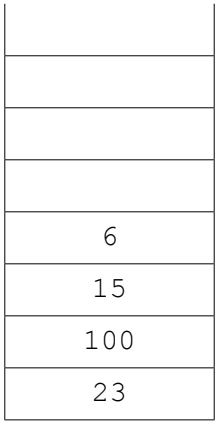
.....

..... **[2]**

5 A computer program stores data input on a stack named `dataItems`. The stack has two sub-programs to add and remove data items from the stack. The stack is implemented as a 1D array, `dataArray`.

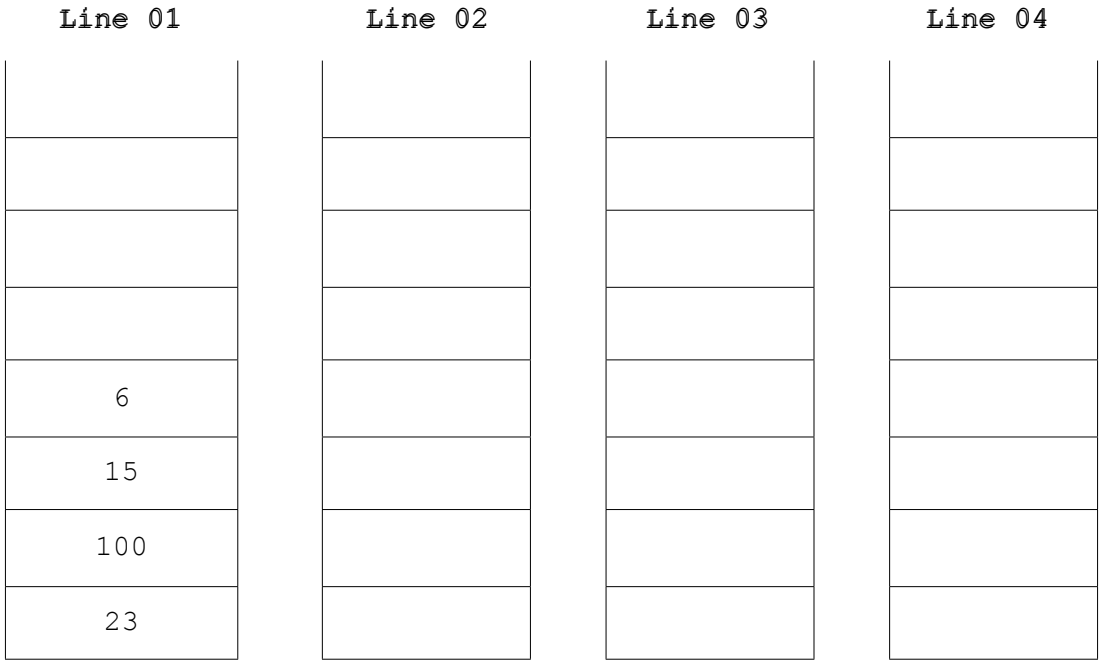
Sub-program	Description
`push()`	The parameter is added to the top of the stack
`pop()`	The element at the top of the stack is removed

The current contents of `dataItems` are shown:



(a) Show the contents of the stack `dataItems` after each line of the following lines of code are run

```
01  push(13)
02  pop()
03  push(10)
04  push(20)
```



[4]

(b) The main program asks a user to push or pop an item from the stack. If the user chooses 'push', the data item is added to the stack. If the user chooses "pop", the next item is removed from the stack, multiplied by 3 and output.

The main program is shown:

```
01 userAnswer = input("Would you like to push or pop an item?")
02 if userAnswer == "push" then
03     push(input("Enter data item"))
04 else
05     print(pop() * 3)
06 endif
```

(i) Before the sub-programs, `push()` and `pop()`, can add or remove items from the stack, a selection statement is used to decide if each action is possible.

Describe the decision that needs to be made in each sub-program and how this impacts the next process.

`push()` .....

.....

.....

.....

`pop()` .....

.....

.....

.....

[4]

(ii) The algorithm does not work when the user enters `"PUSH"` or `"Push"`. The algorithm needs to be changed in order to accept these inputs.

Identify the line number to be changed and state the change that should be made.

Line number .....

Change.....

.....

[2]

(c) The stack is implemented as a 1D array, `dataArray`.

Describe how a 1D array can be set up and used to push and pop items as a stack.

.....

.....

.....

.....

.....

..... **[3]**

(d) As an array, the data in `dataArray` is sorted and then searched for a specific value.

(i) The data in `dataArray` is sorted into ascending order using an insertion sort.

The current contents of `dataArray` are shown:

100	22	5	36	999	12

Show the steps of an insertion sort on the current contents of the array `dataArray`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(ii) The array `dataArray` can now be searched using a binary search.

Describe the stages of a binary search on an array of size `n`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[7]**

(iii) The array has 50 items.

The function, `searchItem()`, performs a linear search for a data item.

```
function searchItem(dataItem)
    for count = 0 to 49
        if dataArray[count] == dataItem then
            return(count)
        endif
    next count
    return(-1)
endfunction
```

Rewrite the function using a while loop.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[4]**

Section B

Answer **all** questions.

6 Kamran is writing a program to manipulate the data for a set of items.

For each item, the program needs to store:
- Item name (e.g. Box)
- Cost (e.g. 22.58)
- Date of arrival (e.g. 1/5/2018)
- Transferred (e.g. true)

The items are added to a queue for processing.

The queue is defined as a class, `itemQueue`.

itemQueue

`theItems[10] : Items`
`head : Integer`
`tail : Integer`
`numItems : Integer`

`constructor`
`enqueuer()`
`dequeuer()`
`setnumItems()`
`getnumItems()`

The `head` attribute points to the first element in the queue. The `tail` attribute points to the next available space in the queue. The `numItems` attribute states how many items are currently in the queue.

(a) The data about the items can be stored using either a record structure, or as objects of a class.

(i) Explain the similarities and differences between a record and a class.

.....

.....

.....

.....

.....

..... **[3]**

(ii) Kamran chooses to use a record structure to store the data about the items.

Record structures may be declared using the following syntax:

```
recordStructure recordstructurename
    fieldname : datatype
    …
endRecordStructure
```

Complete the pseudocode to declare a record called `items`.

```
recordStructure .....

    itemName : .....

    .....: Currency

    .....: Date

    transferred : .....

endRecordStructure
```

[5]

(iii) New records may be created using the following syntax:

```
recordidentifier : recordstructurename
recordidentifier.fieldname = data
…
```

Write a programming statement to create a new item, using the identifier ‘box1’, with the item name “Box”, the cost 22.58, date of arrival 1/5/2018 and transferred true.

.....

.....

.....

.....

.....

.....

.....

..... **[3]**

(b) The array, `theItems`, stores the items in the queue. When the tail of the queue exceeds the last element in the array, it adds a new item to the first element if it is vacant.

For example, in the following queue, the next item to be added would be placed at index 0.

Index	0	1	2	3	4	5	6	7	8	9
Element				Data	Data	Data	Data	Data	Data	Data

(i) Define the term 'queue'.

.....

.....

.....

..... **[2]**

(ii) The attributes in `itemQueue` are all declared as private.

Explain how a private attribute improves the integrity of the data.

.....

.....

.....

..... **[2]**

(iii) The constructor method creates a new instance of `itemQueue` and sets the `head`, `tail` and `numItems` attributes to 0.

Write an algorithm, using pseudocode or program code, for the constructor including the initialisation for all attributes.

.....

.....

.....

.....

.....

..... **[2]**

(iv) The `enqueue` method:

- takes as a parameter the item to insert in the queue
- checks if the queue is full
- reports an error and returns `false` if the queue is full
- does the following if the queue is not full:
  - adds the item to the array at the tail position and adjusts the pointer(s)
  - returns `true`

The attribute `numItems` stores the number of items currently in the queue.

Write an algorithm, using pseudocode or program code, for the `enqueue` method.

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

....

.... **[6]**

(v) Write a programming statement to declare an instance of `itemQueue` called `myItems`.

.....

..... **[1]**

(vi) Write a procedure, `insertItems()`, to ask the user to input the data for an item. The item is then added to the queue `myItems`. The user is continually asked to input data items until the queue is full.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[5]**

(vii) When the main program ends, the items and the queue no longer exist.

Describe how Kamran could amend the program to make sure the items and queue still exist and are used the next time the program is run.

.....

.....

.....

.....

.....

..... **[2]**

(c)* Kamran wants to expand the program to allow it to handle up to 100,000,000 items and to allow him to search for data about items. Kamran is worried that the increase in the number of items will cause a decrease in the performance of the program. He decides to investigate the benefits of caching and concurrent processing.

Evaluate the use of caching and concurrent processing in this scenario and make a recommendation to Kamran.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[9]**

END OF QUESTION PAPER

PLEASE DO NOT WRITE ON THIS PAGE