OCR
Oxford Cambridge and RSA

Friday 24 May 2019 – Morning

AS Level Computer Science

H046/02 Algorithms and problem solving

Time allowed: 1 hour 15 minutes

Do not use:
- a calculator

Please write clearly in black ink. **Do not write in the barcodes.**

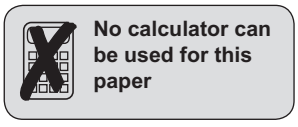
Centre number | | | | | | Candidate number | | | | |

First name(s) _____

Last name _____

INSTRUCTIONS
- Use black ink.
- 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).

INFORMATION
- The total mark for this paper is **70**.
- 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 **16** pages.



1 Janet is designing a piece of software for a furniture company.

The software will allow a user to plan the position of furniture in a room. Users will be able to set the size and shape of a room, and then choose furniture from a library of furniture items. These pieces of furniture will have set sizes and designs and the user will be able to view the room in 3D to see how it looks from a variety of angles.

(a) Janet is using computational thinking techniques during the design process.

(i) Janet is removing some aspects during the design of the software to simplify it and to make it easier to produce.

State the name of the computational thinking technique that Janet is using.

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

(ii) The computational thinking technique in **part (a)(i)** makes it easier to produce the software.

Identify **one** additional reason why this technique is necessary.

.....

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

(iii) Explain, with examples, **two** ways in which Janet will apply the computational thinking technique in **part (a)(i)** to this project.

1 .....

.....

.....

.....

2 .....

.....

.....

.....

[4]

(b) Janet is planning the inputs and outputs for the software.

(i) Identify **two** inputs that the software will need to take.

1 .....

2 .....

[2]

(ii) Identify **two** outputs that the software will need to produce.

1 .....

2 .....

[2]

(c) Janet is going to decompose the problem to produce a set of subprograms.

Explain the benefits of using subprograms to produce this software.

.....

.....

.....

.....

.....

.....

.....

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

(d) The program allows the user to enter dimensions of the room and the furniture. There are preconditions that must be met before the software will draw the room and furniture.

Suggest **two** preconditions that must be met before the software will run.

1 .....

.....

2 .....

.....

[2]

(e)* Janet is planning the testing strategy for the software.

Discuss the different testing methods available to Janet, and make a recommendation for which ones she should use.

.... **[9]**

2 A procedure is shown in the following pseudocode.

The arrays that are passed to the procedure store integer values.

`length` returns the total number of elements the array can hold.

```
01 procedure calculateOnce(data[]:byRef, nextData[]:byRef)
02       if data.length > nextData.length then
03                 loopCount = nextData.length - 1
04       else
05                 loopCount = Data.length - 1
06       endif
07       count = 0
08       while count <= loopCount
09                 data[count] = data[count] + nextData[count]
10                 count = count + 1
11       endwhile
12 endprocedure
```

(a) A decision is made on line `02`.

(i) Identify the line where the second decision is made.

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

(ii) Explain the purpose of the code in lines `02` to `06`.

.....

.....

.....

.....

.....

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

(b) The procedure has parameters passed by reference.

(i) Give the identifiers of the **two** parameters.

1 .....

2 .....

[2]

(ii) State the effect of the array `data[]` being passed by reference and not by value.

.....

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

(c) The program needs a second procedure, `sortData`. It will be called taking the array `data[]` as a parameter by reference.

The procedure will then perform a bubble sort on the data in the array.

(i) Show each stage of a bubble sort on the following contents of `data[]`:

95	10	5	33	100	77	45

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

(ii) Write, using pseudocode, the procedure `sortData`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... **[8]**

(iii) An alternative sorting method is the insertion sort.

Show how an insertion sort will sort the data in the following array.

95	10	5	33	100	77	45

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

3 The current contents of a queue, `colours`, implemented in an array is shown in Fig. 3.1.

red	yellow	green	blue	grey			

```
front = 0

end = 4
```

Fig. 3.1

(a) Describe the purpose of `front` and `end`.

.....

.....

.....

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

(b) The queue has the subprograms `enqueue` and `dequeue`. The subprogram `enqueue` is used to add items to the queue and the subprogram `dequeue` removes items from the queue.

(i) Use the following diagram to show the queue shown in Fig. 3.1 after the following program statements have run:

```
enqueue("orange")
dequeue()
enqueue("maroon")
dequeue()
dequeue()
```


`front = ……`

`end = ……`

[4]

(ii) `enqueue` and `dequeue` are both functions.

State the difference between a procedure and a function.

.....

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

(iii) Describe the steps involved in the `enqueue` algorithm.

.....

.....

.....

.....

.....

.....

.....

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

4 A program corrects the grammar in a line of text. The text is read in from a text file.

(a) The function, `getText`, needs to:

- take the file name as a parameter
- open the file
- read the line of data in the text file into one string
- return the string of data.

Write the function `getText`.

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

(b) The procedure, `fullStop`, needs to:

- ask for a file name as input
- read the data from the file using the function `getText`
- replace the first letter after each full stop with a capital letter if it is currently lower case (if the next character is a space, it must check each successive character until it finds a letter)
- write the edited data back to the text file.

You can assume the text file only contains upper and lower case letters, spaces and full stops.

Part of the ASCII table has been provided:

ASCII Value	Character
65	"A"
90	"Z"
97	"a"
122	"z"
32	" " (space)
46	"." (full stop)

The following functions may be used in your answer:

`asc(`*`character`*`)` returns the ASCII value for a single character, e.g. `asc("A")` would return 65.

`upper(`*`character`*`)` returns the single character in upper case, e.g. `upper("a")` would return “A”.

Write the procedure `fullStop`. **[7]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

END OF QUESTION PAPER

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

BLANK PAGE

PLEASE DO NOT WRITE ON THIS PAGE

PLEASE DO NOT WRITE ON THIS PAGE