Oxford Cambridge and RSA

Tuesday 21 May 2024 – Afternoon

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.
- 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 **70**.
- The marks for each question are shown in brackets **[ ]**.
- Quality of extended response will be assessed in questions marked with an asterisk (*).
- This document has **16** pages.

ADVICE
- Read each question carefully before you start your answer.

© OCR 2024 [601/5030/0]
DC (ST) 319817/6
OCR is an exempt Charity

Answer **all** the questions.

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.

(a) Eve would like to create a computerised version of this game.

(i) She has been told that she should make use of abstraction when creating the game.

Describe what is meant by the term ‘abstraction’.

.....

.....

.....

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

(ii) Give **three** examples of how Eve could use abstraction when creating her game.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(iii) Give **two** reasons why Eve should use abstraction when designing the game.

1 .....

.....

2 .....

.....

[2]

© OCR 2024

(b) Eve would like to break the problem down into smaller sub problems so that each sub problem will complete one specific task.

Identify **three** sub problems that Eve can use in her game.

1 .....

.....

2 .....

.....

3 .....

.....

[3]

(c) 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]

© OCR 2024

2 A programmer is designing a program that will store data.

The programmer is deciding whether to store the data in a stack or a queue.

(a) Identify **one** similarity and **one** difference between a stack and a queue.

Similarity .....

.....

Difference .....

.....
[2]

(b) The pseudocode function, `enqueue`, inserts an item into a queue.

```
function enqueue(item)
        if tailPointer >= queue.length then
              return false
        else
              queue[tailPointer] = item
              tailPointer = tailPointer + 1
              return true
        endif
endfunction
```

(i) Give the name of the parameter in the function `enqueue`.

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

(ii) Give the name of **one** global variable that is used in the function `enqueue`.

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

© OCR 2024

(iii) Describe **one** benefit and **one** drawback of using global variables instead of parameter passing in a subroutine.

Benefit .....

.....

.....

.....

Drawback .....

.....

.....

.....

[4]

(iv) The function `enqueue` can be called by the main program.

Explain why the function `enqueue` returns true or false values, and how this can be used by the main program that calls the function.

.....

.....

.....

.....

.....

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

© OCR 2024

(v) The pseudocode function, `dequeue`, removes and returns the first item in the queue. If the queue is empty, the function returns the string `"EMPTY"`.

```
01  function dequeue(data)
02          if headPointer != tailPointer then
03               return "EMPTY"
04          elseif
05               value = queue[headPointer]
06               return value
07               headPointer = headPointer + 1
08          endif
09  endfunction
```

The function `dequeue` has **several** errors.

Identify the line number of any **three** errors **and** state the correction required.

Error 1 Line Number .....

Error 1 Correction .....

Error 2 Line Number .....

Error 2 Correction .....

Error 3 Line Number .....

Error 3 Correction .....

[3]

© OCR 2024

(vi) The programmer has corrected all of the errors in the function `dequeue`.

The main program repeatedly calls the function `dequeue` until all of the elements in the queue have been output.

Write the main program using pseudocode or program code.

.....

.....

.....

.....

.....

.....

.....

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

© OCR 2024

3* Kofi and Zac both write a different pseudocode algorithm to read the data from a text file into an array.

Kofi's Algorithm	Zac's Algorithm

`fileName = "data.txt"`
`fileToRead = openRead(fileName)`
`for x = 0 to 1000`
`  anyData = fileToRead.endOfFile()`
`  if NOT anyData then`
`  x = 1001`
`  else`
`  dataValue = fileToRead.readLine()`
`  array[x] = dataValue`
`  endif`
`next x`
`fileToRead.close()`
	`function readData(fileName)`
	`  array data[100]`
	`  x = 0`
	`  fileToRead = openRead(fileName)`
	`  while NOT fileToRead.endOfFile()`
	`  data[x] = fileToRead.readLine()`
	`  x = x + 1`
	`  endwhile`
	`  return data`
	`endfunction`

The solution needs to be used in different programs. Each program will use a different text file where the number lines in the text file is unknown.

Compare the suitability of each algorithm for the given problem.

You should include the following in your answer:
- the suitability of the programming techniques including the use of loops
- how effectively the solution meets the requirements.

[9]

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

© OCR 2024

© OCR 2024

4 Charlie is developing a computer game using a development lifecycle.

(a) Complete the table by describing each method of software development.

Method	Description
Extreme programming	....
Waterfall lifecycle	....
Spiral model	....

[6]

(b) Charlie uses alpha testing.

Describe what is meant by 'alpha testing'.

.....

.....

.....

.....

.....

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

© OCR 2024

5 A program makes use of searching and sorting algorithms.

(a) The following incomplete pseudocode algorithm uses a binary search to find the integer `numberToFind` in the array `array`. It returns the index of the array or –1 if the integer is not found.

Complete the pseudocode algorithm.

```
function binarySearch(array, ........................................ )
     lowerbound = 0
     upperbound = array.length - 1

     while true
          if(upperbound < lowerbound) then
                return ........................................
          else
                mid = (upperbound + lowerbound) ........................................
                if(array[mid] < numberToFind) then
                     lowerbound = mid ........................................
                elseif(array[mid] > numberToFind) then
                     upperbound = mid ........................................
                else
                     return ........................................
                endif
          endif
     endwhile
endfunction
```

[6]

© OCR 2024

(b) An array stores the following data:

20	8	33	16

(i) Describe how the given data will be sorted into **descending** numerical order using an insertion sort.

You should refer to the data in this array throughout your answer.

.....

.....

.....

.....

.....

.....

.....

.....

.....

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

© OCR 2024

(ii) The size of the array has now been increased to **seven** elements.

The insertion sort algorithm needs to be tested to ensure it sorts a range of test data into **descending** numerical order.

For example, the test data in the array here will test to see if the insertion sort will sort data in the opposite order.

1	2	3	4	5	6	7

Give **two other** different sets of data in the array that can be used to test the insertion sort and state the purpose of each set of test data.

Each test needs to have a different purpose.

Set One

Test Data 1

Purpose of test data 1

.....

.....

Set Two

Test Data 2

Purpose of test data 2

.....

.....

[2]

© OCR 2024

6 A program needs writing to allow a user to play the game fizz buzz.

The rules are:
- the user enters the numbers from 1 to 100 in turn
- if the number is exactly divisible by 3 the user enters “fizz” instead of the number
- if the number is exactly divisible by 5 the user enters “buzz” instead of the number
- if the number is exactly divisible by both 3 and 5 the user enters “fizz buzz” instead of the number
- the game will continue even if a wrong answer has been input until 100 is reached.

For example, the first 10 numbers will be:
1
2
fizz
4
buzz
fizz
7
8
fizz
buzz

The program needs to:
- repeatedly allow the user to enter each number, or word(s) one at a time
- output a suitable message if the input is incorrect, telling the user the correct number or correct word. For example, *“Incorrect, the answer is 4”* or *“Incorrect, the answer is fizz buzz”*.

Write this algorithm using pseudocode or program code. **[8]**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

© OCR 2024

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

END OF QUESTION PAPER

© OCR 2024

EXTRA ANSWER SPACE

If you need extra space use this lined page. You must write the question numbers clearly in the margin.

OCR
Oxford Cambridge and RSA

Copyright Information

OCR is committed to seeking permission to reproduce all third-party content that it uses in its assessment materials. OCR has attempted to identify and contact all copyright holders whose work is used in this paper. To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced in the OCR Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download from our public website (www.ocr.org.uk) after the live examination series.

If OCR has unwittingly failed to correctly acknowledge or clear any third-party content in this assessment material, OCR will be happy to correct its mistake at the earliest possible opportunity.

For queries or further information please contact The OCR Copyright Team, The Triangle Building, Shaftesbury Road, Cambridge CB2 8EA.

OCR is part of Cambridge University Press & Assessment, which is itself a department of the University of Cambridge.

© OCR 2024