A Level · OCR · Computer Science

51 mins 8 questions

Exam Questions

7.1 Programming Techniques

Data Types / Arithmetic, Logical & Boolean Operators / Programming Constructs / Selection / Iteration / Modularity, Functions & Procedures / Parameter Passing / Recursion / Global & Local Variables / Integrated Development Environment (IDE) / Programming Classes, Objects, Methods & Attributes / Programming Inheritance / Programming Encapsulation / Programming Polymorphism

Scan here to return to the course
or visit savemyexams.com

Total Marks /51

1 A programmer has designed a program that includes a reusable program component

The reusable program component is a function called isInteger(). This will take a string as an argument and then check that each digit is between 0 and 9. For example if 103 is input, it will check that the digits 1, 0 and 3 are each between 0 and 9.

The asc() function returns the ASCII value of each digit. For example asc("1") returns 49.

The ASCII value for 0 is 48. The ASCII value for 9 is 57.

```
01 function isInteger(number)
02   result = true
03   for count = 0 to number.length-1
04     asciiValue = asc(number.substring(count, 1))
05     if not(asciiValue >= 48 and asciiValue <= 57) then
06       result = false
07     endif
08   next count
09   return result
10 endfunction
```

Give the line number where the iteration construct starts in the function isInteger().

.....

(1 mark)

2 A card game uses a set of 52 standard playing cards. There are four suits; hearts, diamonds, clubs and spades. Each suit has a card with a number from; 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13.

The card game randomly gives 2 players 7 cards each. The unallocated cards become known as the deck.

The players then take it in turns to turn over a card. A valid move is a card of the same suit or the same number as the last card played.

The winner is the first player to play all of their cards.

A function, checkValid(), takes the card the player has selected, and the last card played as parameters.

It returns true if the player's move is valid and returns false if the player's move is not valid.

State the reason why checkValid() is a function and not a procedure.

.....

(1 mark)

3 The following pseudocode procedure performs an insertion sort on the array parameter.

```
01 procedure insertionSort(dataArray:byRef)
02   for i = 1 to dataArray.Length - 1
03     temp = dataArray[i]
04     tempPos = i – 1
05     exit = false
06     while tempPos >= 0 and exit == false
07       if dataArray[tempPos] < temp then
08         dataArray[tempPos + 1] = dataArray[tempPos]
09         tempPos = tempPos - 1
10       else
11         exit = true
12       endif
13     endwhile
14   dataArray[tempPos + 1] = temp
15   next i
16 endprocedure
```

Explain why dataArray is passed by reference and not by value.

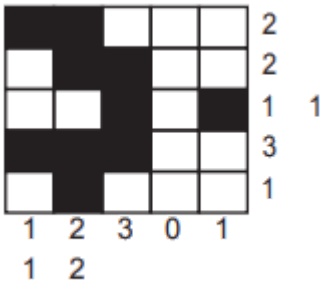
.....

.....

(2 marks)

4 (a) 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 black 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.



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 black and if it is black 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.

Juan creates 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 black box.

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().

```
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
```

(5 marks)

(b) **Explain** the purpose of line 03 in the function countRow.

(2 marks)

(c) **Describe** the purpose of branching and iteration in the function countRow.

(3 marks)

(d) 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 a** 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
```

Write pseudocode or program code for the procedure displayRowAnswer()

(6 marks)

(e) Juan passed the two arrays as parameters, but he did consider making them globally accessible.

Compare the use of global and local variables and data structures in this program. Include the use of parameters and program efficiency in your answer.

(9 marks)

5 Hugh has written a recursive function called thisFunction() using pseudocode.

```
01 function thisFunction(theArray, num1, num2, num3)
02   result = num1 + ((num2 - num1) DIV 2)
03   if num2 < num1 then
04     return -1
05   else
06     if theArray[result] < num3 then
07       return thisFunction(theArray, result + 1, num2, num3)
08     elseif theArray[result] > num3 then
09       return thisFunction(theArray, num1, result - 1, num3)
10     else
11       return result
12     endif
13   endif
14 endfunction
```

The function DIV calculates integer division, e.g. 5 DIV 3 = 1

Hugh could have written thisFunction() using iteration instead of recursion.

Compare two differences between recursion and iteration.

.....

.....

.....

.....

(4 marks)

6 A class, Board, is used to store the 10 row (x coordinate) by 20 column (y coordinate) grid

The design for the Board class, its attributes and methods is shown here.

class: Board
attributes: private grid : Array of Treasure
methods: new() function getGridItem(x, y) function setGridItem(x, y, treasureToInsert)

The constructor initialises each space in the grid to a treasure object with value as -1 and level as an empty string.

Complete the following pseudocode for the constructor method.

```
public procedure new()

 for row = ..... to 9

  for column = 0 to .....

   ..... [row, column] = new Treasure(.....,"")

  next .....

 next row

endprocedure
```

(5 marks)

7 A text-based computer game allows a user to dig for treasure on an island. The island is designed as a grid with 10 rows and 20 columns to store the treasure. Each square is given an x and y coordinate. Some of the squares in the grid store the name of a treasure object. Each treasure object has a value, e.g. 100 and a level, e.g. "Bronze."

A procedure, guessGrid():

- takes a Board object as a parameter
- accepts the row (x) and column (y) coordinates from the user
- outputs "No treasure" if there is no treasure found at the coordinate (level is an empty string)
- if there is treasure at that coordinate, it outputs the level and the value of the treasure in an appropriate message.

Write the procedure guessGrid() using either pseudocode or program code.

(7 marks)

8 The recursive function thisFunction().

```
01 function thisFunction(theArray, num1, num2, num3)
02   result = num1 + ((num2 - num1) DIV 2)
03   if num2 < num1 then
04     return -1
05   else
06     if theArray[result] < num3 then
07       return thisFunction(theArray, result + 1, num2, num3)
08     elseif theArray[result] > num3 then
09       return thisFunction(theArray, num1, result - 1, num3)
10     else
11       return result
12     endif
13   endif
14 endfunction
```

Rewrite the function thisFunction() so that it uses iteration instead of recursion.

You **should** write your answer using pseudocode or program code.

(6 marks)