**COMSATS University Islamabad, Islamabad campus**

**Department of Computer Science**

**Terminal Examination (Subjective Part), Fall 2020**

**Class:** BCS/BSE 2  **Marks:** 30

**Subject:** CSC103 Programming Fundamentals **Time:** 02 hrs.

**Instructors:** Dr. Manzoor / Mr. Rizwan / Ms. Saadia / Dr. Behjat **Dated:** January 11, 2021

**Question 01 – Recursion Points: 06**

Take two coding problems that can be solved recursively. You are required to provide their:

* 1. Iterative implementation
  2. Recursive implementation
  3. Call stack information of recursive implementation (you can draw call stacks for each problem and insert pictures in the word file)

**Example 1:**

|  |
| --- |
| Iterative implementation  public class Q1Iter {  public static void main(String[] args) {  int N = **2,** i = **1;** for(i=**1;**i<**6;**i++){  System.*out*.println(N + " \* " + i + " = " + N \* i)**;** }  } } |
| Recursive implementation  public class Question1 {  public static void table(int N**,** int i) {  if(i> **5**)  return**;** System.*out*.println(N + " \* " + i + " = " + N \* i)**;** *table*(N**,** i+**1**)**;** }  public static void main(String[] args) {  int N = **2,**i = **1;** *table*(N**,**i)**;** } } |
| Call stack  A picture containing text, whiteboard  Description automatically generated |

**Example 2:**

|  |
| --- |
| Iterative implementation  public class Q1Ex2It {  public static void main(String args[]){  int i**,**fact=**1;** int number=**3;** for(i=**1;**i<=number**;**i++){  fact=fact\*i**;** }  System.*out*.println("Factorial of "+number+" is: "+fact)**;** } } |
| Recursive implementation  public class Q1Ex2 {  static int factorial(int n){  if (n == **0**)  return **1;** else  return(n \* *factorial*(n-**1**))**;** }  public static void main(String[] args) {  int i**,**fact=**1;** int number=**3;** fact = *factorial*(number)**;** System.*out*.println("Factorial of "+number+" is: "+fact)**;** } } |
| Call stack  Diagram, schematic  Description automatically generated |

**Question 02 – Two-dimensional array Points: 10**

Code an application that displays the number of times a value appears in a two-dimensional array. It displays the number of times each of the numbers from 1 through 9 appears in the numbers array.

**Example**: Suppose the user entered the following array.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 7 |
| 2 | 2 | 2 |
| 1 | 1 | 7 |

**Output**: The value 1 appears 3 times.

The value 2 appears 4 times.

The value 7 appears 2 times.

|  |
| --- |
| import java.util.Scanner**;** public class Question\_2 {  public static int num\_check(int[][] num\_arr**,**int current\_Num**,**int row**,**int col) {  int count = **0;** for(int i=**0;** i<row**;** i++){  for(int j=**0;**j<col**;**j++){  if(current\_Num == num\_arr[i][j]){  count++**;** }  }    }  return count**;** }  public static void main(String[] args) {  Scanner scan = new Scanner(System.*in*)**;** int row**,**col**,**count=**0;** System.*out*.println("Enter number of rows: ")**;** row = scan.nextInt()**;** System.*out*.println("Enter number of columns: ")**;** col = scan.nextInt()**;** int[][] num = new int[row][col]**;** for(int i=**0;** i<row**;** i++){  for(int j=**0;**j<col**;**j++){  System.*out*.println("Enter values: ")**;** num[i][j] = scan.nextInt()**;** }  }   for(int i=**0;** i<row**;** i++){  for(int j=**0;**j<col**;**j++){  System.*out*.print(num[i][j])**;** }  System.*out*.println()**;** }  int cur\_Num = num[**0**][**0**]**;** int temp\_num = cur\_Num**;** for(int i=**0;** i<row**;** i++){  for(int j=**0;**j<col**;**j++){  cur\_Num = num[i][j]**;** count = *num\_check*(num**,**cur\_Num**,**row**,**col)**;** System.*out*.println(cur\_Num+ " comes " + count+ " times")**;** }    }   } } |

**Question 03 – Strings Points: 04**

Given a string, determine the length of the widest/biggest fragment, where all the characters of the fragment are same.

**Example:** a**bbbb**aca**ccc**a**fff**d

**Output:** 4

|  |
| --- |
| import java.util.Scanner**;** public class Q3 {  public static void main(String[] args) {  Scanner scan = new Scanner(System.*in*)**;** String alphabets = " "**;** System.*out*.println("ENter String which you want to check Without space: ")**;** alphabets = scan.next()**;** int largest\_frag\_total = **0;** int widest\_Frag = **0;** for(int i = **0;**i<alphabets.length()**;**i++){  for(int j=**0;**j<alphabets.length()**;** j++){  if(alphabets.charAt(i) == alphabets.charAt(j))  largest\_frag\_total++**;** else{  if(widest\_Frag<largest\_frag\_total)  widest\_Frag = largest\_frag\_total**;** largest\_frag\_total = **0;** }  }  }  System.*out*.println("Largest Fragment length is: " + widest\_Frag)**;** } } |

**Question 04 – Text I/O Points: 10**

Write a program to create a file named **randomNumbers.txt** if it does not exist. Write 25 numbers, generated randomly, to the file using text I/O. The numbers must be separated by spaces in the file. Read the data back from the file and display the data in decreasing order.

|  |
| --- |
| import java.io.File**;** import java.io.IOException**;** import java.io.PrintWriter**;** import java.sql.Array**;** import java.util.Random**;** import java.util.Scanner**;** public class Question4 {  public static int[] read() {  File file = new File("randomNumbers.txt")**;** String Data = " "**;** int[] arr = new int[**25**]**;** int i = **0;** try {  Scanner readFile = new Scanner(file)**;** while(readFile.hasNext()){  Data = readFile.next()**;** arr[i] = Integer.*parseInt*(Data)**;** i++**;** }   }  catch (IOException e){   }  return arr**;** }  public static void write() {  Random rand = new Random()**;** File file = new File("randomNumbers.txt")**;** try{  PrintWriter fileWriter = new PrintWriter("randomNumbers.txt")**;** for(int i = **0;** i< **25;** i++){  fileWriter.write(rand.nextInt(**25**)+ " ")**;** fileWriter.flush()**;** }     }  catch (IOException e){   }    }  public static void main(String[] args) {  int[] arr = new int[**25**]**;** *write*()**;** arr = *read*()**;** int size = arr.length**;** for (int i = **1;** i < size**;** i++) {  int key = arr[i]**;** int j = i - **1;** while (j >= **0** && key > arr[j]) {   arr[j + **1**] = arr[j]**;** --j**;** }  arr[j + **1**] = key**;** }   for(int i = **0;**i<**25;**i++){  System.*out*.println(arr[i])**;** }    } } |