**LAB assement 1 soft-record**

**CSE1007: Java Programming**

Programme: B.Tech ; Branch: ECE

Class number: VL2023240105377

Slot: L39+L40

Date: 08-08-2023 Time: 3:50-5:30

VENUE: SJTG17

Reg No: 20BEC0512

Name: B. Karthik reddy



**School of Computer Science and Engineering**

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-1** |
| 08-08-2023 | **Check whether a given number is palindrome or not** |
|  | |
| Program | import java.util.\*;  public class palindrome {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("Enter the string:");  String s1 = sc.nextLine();  String s2 = "";  int n = s1.length();  for (int i = n - 1; i >= 0; i--) {  s2 = s2 + s1.charAt(i);  }  if (s1.equals(s2)) {  System.out.println("Hurray! word " + s1 + " is a palindrome.");  } else {  System.out.println("Oops! word " + s1 + " is not a palindrome.");  }  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-2** |
| 08-08-2023 | **Generate prime numbers from 1 to n** |
|  | |
| Program | import java.util.\*;  public class primenumbers {  public static void main(String[] args) {  // user input  Scanner sc = new Scanner(System.in);  System.out.println("Enter the number N for showing the prime numbers which are from 1 To N :");  int n = sc.nextInt();  System.out.println("The prime numbers are :");  for (int i = 0; i <= n; i++) {  // checking number is prime or not by calling isprime function  if (isprime(i)) {  System.out.print(i + " ");  }  }  sc.close();  }  // checks if the number is divisible any other number except 1 and itself and  // returns a boolean value  static boolean isprime(int k) {  if (k < 2) {  return false;  }  for (int j = 2; j < k; j++) {  if (k % j == 0) {  return false;  }  }  return true;  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-3** |
| 08-08-2023 | **Find the kth smallest element in an array** |
|  | |
| Program | import java.util.\*;  public class Kthsmallest {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  // size of the array:  System.out.println("enter the size of the array: ");  int n = sc.nextInt();  int[] arr = new int[n];  // assigning elements:  for (int i = 0; i < n; i++) {  System.out.println("Enter the " + (i + 1) + " value:");  arr[i] = sc.nextInt();  }  // asking the Kth value from the user:  System.out.println("enter the Kth value:");  int k = sc.nextInt();  // sorting the array using bubble sort:  for (int i = 0; i < n - 1; i++) {  boolean swapped = false;  for (int j = 0; j < n - i - 1; j++) {  if (arr[j] > arr[j + 1]) {  swapped = true;  int temp = arr[j];  arr[j] = arr[j + 1];  arr[j + 1] = temp;  }  }  if (!swapped) {  break;  }  }  // printing the kth smallest value from the sorted array:  if (k <= n) {  System.out.println("the Kth smallest element in the array is:" + arr[k - 1]);  } else {  System.out.println("Kth value is greater than the Array size");  }  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-4** |
| 08-08-2023 | **Merge two 1D arrays into one 1D array excluding the repeated elements** |
|  | |
| Program | import java.util.\*;  public class mergingarrays {  public static boolean check(int[] arr, int k) {  boolean check = false;  for (int i = 0; i < arr.length; i++) {  if (arr[i] == k) {  check = true;  }  }  return check;  }  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  // taking the size of the arrays from the user:  System.out.println("enter the size of the 1st array: ");  int n1 = sc.nextInt();  System.out.println("enter the size of the 2nd array: ");  int n2 = sc.nextInt();  // intializing and decalaration of 1st and 2nd array:  int[] arr1 = new int[n1];  int[] arr2 = new int[n2];  int[] arr3 = new int[n1 + n2];  // assignment of the arrays:  for (int i = 0; i < n1; i++) {  System.out.println("Enter the " + (i + 1) + " value of 1st array :");  arr1[i] = sc.nextInt();  }  for (int i = 0; i < n2; i++) {  System.out.println("Enter the " + (i + 1) + " value of 2nd array :");  arr2[i] = sc.nextInt();  }  // adding all the elements of the 1st array and then adding only unique elements  // of 2nd array:  for (int i = 0; i < n1; i++) {  arr3[i] = arr1[i];  }  // now adding unique elements from 2nd array:  int k = n1;  for (int i = 0; i < n2; i++) {  boolean found = check(arr3, arr2[i]);  if (!found) {  arr3[k] = arr2[i];  k++;  }  }  // displaying the combined array elements:  System.out.println("the combined array of array 1 and 2 with removing the duplicates is : ");  for (int i = 0; i < arr3.length; i++) {  if (arr3[i] != 0) {  System.out.print(arr3[i] + " ");  }  }  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-5** |
| 08-08-2023 | **Display the sum of perfect numbers in a 2D array** |
|  | |
| Program | import java.util.\*;  public class perfectnumbers {  public static boolean perfectnumber(int x) {  boolean perfectnumber = false;  int count = 0;  for (int i = 1; i < x; i++) {  if (x % i == 0) {  count = count + i;  }  }  if (count == x) {  perfectnumber = true;  }  return perfectnumber;  }  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  // taking the number of rows values from the user:  System.out.println("enter the number of rows: ");  int x = sc.nextInt();  System.out.println("enter the number of columns: ");  int y = sc.nextInt();  int[][] arr = new int[x][y];  // assigning the elements and implementing the logic too:  int sum = 0;  for (int i = 0; i < x; i++) {  for (int j = 0; j < y; j++) {  System.out.println("enter the value of " + (i + 1) + " row and " + (j + 1) + " element");  arr[i][j] = sc.nextInt();  if (perfectnumber(arr[i][j])) {  sum = sum + arr[i][j];  }  }  }  System.out.println("the sum of the perfect numbers in the 2-D array is: " + sum);  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-6** |
| 08-08-2023 | **Write a program to extract a portion of a character string and print the extracted string. Assume that m characters are extracted, starting with the nth character** |
|  | |
| Program | import java.util.\*;  public class substring {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);  System.out.println("enter the string: ");  // reading the string from the user:  String s = sc.nextLine();  System.out.println("enter the Starting substring value: ");  // reading the starting substring value :  int x = sc.nextInt();  System.out.println("enter the ending substring value: ");  // reading the ending substring value:  int y = sc.nextInt();  System.out.println("the substring is : ");  for (int i = x; i <= y; i++) {  System.out.print(s.charAt(i));  }  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |

|  |  |
| --- | --- |
| Assessment:  Date: | **Problem Statement-7** |
| 08-08-2023 | **Display the last occurrence of a character in a string** |
|  | |
| Program | import java.util.\*;  public class lastoccurence {  public static void main(String[] args) {  int result = 0;  Scanner sc = new Scanner(System.in);  System.out.println("enter the String:");  String s = sc.nextLine();  System.out.println("enter the character to be searched for:");  char c = sc.next().charAt(0);  for (int i = 0; i < s.length(); i++) {  if (s.charAt(i) == c) {  result = i;  }  }  System.out.println("the last occurence of the given character in the string was : " + result);  sc.close();  }  } |
|  | |
| Code Screenshot |  |
| Output 1: |  |
| Output 2: |  |
|  | |
| Observations |  |
|  | |
| References |  |