**C-DAC Mumbai Date 26/09/2024**

**Subject: Algorithm and Data Structure**

**Assignment 1**

**Solve the assignment with following thing to be added in each question.**

-Program

-Flow chart

-Explanation

-Output

-Time and Space complexity

1. Printing Patterns

Problem: Write a Java program to print patterns such as a right triangle of stars.

Test Cases:

Input: n = 3

Output:

\*

\*\*

\*\*\*

Input: n = 5

Output:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Code : **package** org.example.in;

**import** java.util.Scanner;

**public** **class** Star\_Pattern {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the numbers");

**int** n = sc.nextInt();

**for**(**int** i =1; i<=n; i ++) {

**for** (**int** j =1; j<=i; j++) {

System.***out***.print("\*");

}

System.***out***.println();

}

}

}

Output : A screenshot of a computer

Description automatically generated

2. Remove Array Duplicates

Problem: Write a Java program to remove duplicates from a sorted array and return the new length of the array.

Test Cases:

Input: arr = [1, 1, 2]

Output: 2

Input: arr = [0, 0, 1, 1, 2, 2, 3, 3]

Output: 4

**package** org.example.in.a;

**import** java.util.Scanner;

//import java.util.Scanner;

**public** **class** RemoveDuplicates {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the numbers");

**int** size = sc.nextInt();

**int** [] a = **new** **int**[size];

/\*int [] temp = new int[a.length];

int j = 0;\*/

System.***out***.println("Enter the sorted elements");

**for** (**int** i = 0; i < size; i++) {

a[i] = sc.nextInt();

}

// To store temp results

**int** [] temp = **new** **int**[a.length];

**int** j = 0;

**for** (**int** i=0; i<a.length-1; i++) {

**if**(a[i]!= a[i+1]) {

temp[j] = a[i];

j++;

}

}

// Adding the last element of the original array to temp array

temp[j] = a[a.length-1];

j++;

**for** (**int** i =0 ;i<j; i ++) {

System.***out***.print(temp[i]+ " ");

}

System.***out***.println("\nLength of array after removing duplicates: " + j);

sc.close();

}

}

Output: A screenshot of a computer

Description automatically generated

3. Remove White Spaces from String

Problem: Write a Java program to remove all white spaces from a given string.

Test Cases:

Input: "Hello World"

Output: "HelloWorld"

Input: " Java Programming "

Output: "JavaProgramming"

**package** org.example.in.b;

**import** java.util.Scanner;

//import java.util.Scanner;

**public** **class** RemoveWhiteSpaces {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String input = sc.nextLine();

String result = *removeWhiteSpaces*(input);

System.***out***.println("String after removing white spaces: \"" + result + "\"");

sc.close();

}

**public** **static** String removeWhiteSpaces(String str) {

**return** str.replaceAll("\\s+", "");

}

}

Output : A screenshot of a computer

Description automatically generated

4. Reverse a String

Problem: Write a Java program to reverse a given string.

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

Code :

**package** org.example.in.c;

**import** java.util.Scanner;

//import java.util.Scanner;

**public** **class** ReverseAString {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter a string: ");

String name = sc.nextLine();

String result = *rev*(name);

System.***out***.println("Reverse of " + name + " is " + *rev*(name));

}

**public** **static** String rev(String name) {

**int** len = name.length();

String rev = "";

**for** (**int** i = len-1; i>=0; i--){

rev = rev+name.charAt(i);

}

**return** rev;

}

}

Output:

A screenshot of a computer

Description automatically generated

5. Reverse Array in Place

Problem: Write a Java program to reverse an array in place.

Test Cases:

Input: arr = [1, 2, 3, 4]

Output: [4, 3, 2, 1]

Input: arr = [7, 8, 9]

Output: [9, 8, 7]

Code:

**package** org.example.in.e;

**import** java.util.Scanner;

**class** ReverseArray {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter the size of an array ");

**int** n = sc.nextInt();

**int** [] a = **new** **int**[n];

System.***out***.println("Elements of the arrays are: ");

**for** (**int** i=0; i<n ;i++) {

a[i] = sc.nextInt();

System.***out***.print("--->" + a[i]);

}

System.***out***.println("After reversing the array");

**for** (**int** j=n-1; j>=0; j--) {

System.***out***.println("--->" + a[j]);

}

sc.close();

}

}

Output:

A screenshot of a computer program

Description automatically generated

6. Reverse Words in a String

Problem: Write a Java program to reverse the words in a given sentence.

Test Cases:

Input: "Hello World"

Output: "World Hello"

Input: "Java Programming"

Output: "Programming Java"

Code:

**package** org.example.in.f;

**import** java.util.Scanner;

**class** ReverseWords {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter String : ");

String str=sc.nextLine();

String[] arr=str.split(" ");

String rev="";

**for**(**int** i=arr.length-1;i>=0;i--)

{

rev=rev+arr[i];

rev=rev +" ";

}

System.***out***.println(rev);

}

}

Output:

A screenshot of a computer

Description automatically generated

7. Reverse a Number

Problem: Write a Java program to reverse a given number.

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

Code :

**package** org.example.in.d;

**import** java.util.Scanner;

**public** **class** ReverseNumber {

**public** **static** **void** main(String[] args) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the numbers: ");

**int** num = sc.nextInt();

**int** result = *revnum*(num);

System.***out***.println("Reverse of " + num + " is " + result);

sc.close();

}

**public** **static** **int** revnum(**int** num) {

**int** rev = 0 , rem = 0;

**while**(num !=0) {

rem = num %10;

rev = rev \* 10 + rem;

num = num/10;

}

**return** rev;

}

}

Output:

A screenshot of a computer

Description automatically generated

8. Array Manipulation

Problem: Perform a series of operations to manipulate an array based on range update queries. Each query adds a value to a range of indices.

Test Cases:

Input: n = 5, queries = [[1, 2, 100], [2, 5, 100], [3, 4, 100]]

Output: 200

Input: n = 4, queries = [[1, 3, 50], [2, 4, 70]]

Output: 120

Code:

import java.util.Scanner;

import java.util.Arrays;

class QueryUpdate

{

    public static void main(String[] args)

    {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter Size Of Array: ");

        int size=sc.nextInt();

        int[] arr=new int[size];

        char ch='y';

        while(ch!='n')

        {

            System.out.println("Enter Starting point: ");

            int start=sc.nextInt();

            System.out.println("Enter Ending Point: ");

            int end=sc.nextInt();

            System.out.println("Enter Element");

            int ele=sc.nextInt();

            for(int i=start;i<end;i++)

            {

                arr[i]+=ele;

            }

            System.out.println("More Queries : ");

            ch=sc.next().charAt(0);

        }

        System.out.println(Arrays.toString(arr));

        Arrays.sort(arr);

        System.out.println(arr[size-1]);

    }

}

Output:

A screenshot of a computer

Description automatically generated

9. String Palindrome

Problem: Write a Java program to check if a given string is a palindrome.

Test Cases:

Input: "madam"

Output: true

Input: "hello"

Output: false

Here’s a continuation of the list of assignment questions starting from question 21, with two test cases for each:

Code:

**package** org.example.in.g;

**import** java.util.Scanner;

**class** StringPalindrome {

**public** **static** **void** main(String[] args) {

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter String : ");

String str=sc.nextLine();

String rev="";

**for**(**int** i=str.length()-1;i>=0;i--)

{

rev=rev+str.charAt(i);

}

**if**(str.equals(rev))

{

System.***out***.println("true");

}

**else**

{

System.***out***.println("false");

}

System.***out***.println("Reversed String: "+rev);

}

}

Output:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

10. Array Left Rotation

Problem: Write a Java program to rotate an array to the left by d positions.

Test Cases:

Input: arr = [1, 2, 3, 4, 5], d = 2

Output: [3, 4, 5, 1, 2]

Input: arr = [10, 20, 30, 40], d = 1

Output: [20, 30, 40, 10]

Code:

import java.util.Scanner;

import java.util.Arrays;

class ArrLeftRotation

{

    public static void main(String[] args)

    {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter Size: ");

        int size=sc.nextInt();

        int[] arr=new int[size];

        System.out.println("Enter Number of rotations: ");

        int rotation=sc.nextInt();

        int[] temparr=new int[rotation];

        for(int i=0;i<arr.length;i++)

        {

            System.out.println("Enter Value for arr["+i+"] : ");

            arr[i]=sc.nextInt();

            if(i<rotation)

                temparr[i]=arr[i];

        }

        System.out.println("Original array: "+Arrays.toString(arr));

        System.out.println("Temp array: "+Arrays.toString(temparr));

        int j=0;

        for(int i=0;i<arr.length;i++)

        {

            if(i+rotation<=arr.length-1)

                arr[i]=arr[i+rotation];

            else

                arr[i]=temparr[j++];

        }

        System.out.println("Output array: "+Arrays.toString(arr));

    }

}

Output:

