**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:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

/\*Pattern right triangle of stars\*/

import java.util.Scanner;

class Pattern1{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

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

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

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

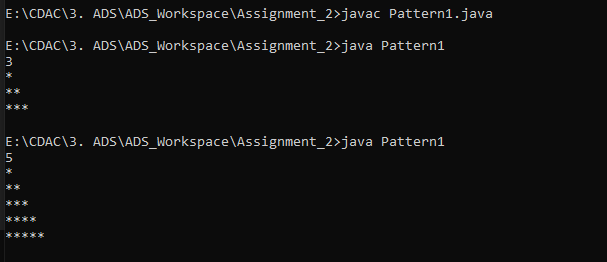
}

System.out.println();

}

}

}



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

/\* Remove duplicates from a sorted array \*/

import java.util.Scanner;

class RemoveDuplicates{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int size =sc.nextInt();

int[] arr = new int[size];

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

arr[i] = sc.nextInt();

}

int count = 0;

for(int i=1; i<arr.length; i++){

if(arr[i] != arr[count]){

count++;

arr[count] = arr[i];

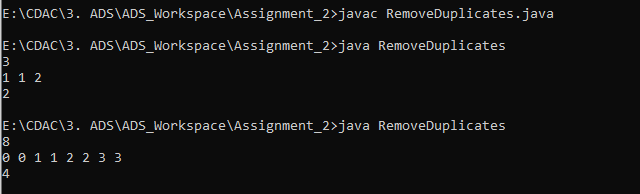
}

}

System.out.println(count+1);

}

}



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"

/\* Remove all white spaces from a given string.\*/

import java.util.Scanner;

class RemoveSpacesOfString{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

StringBuilder str1 = new StringBuilder();

for(int i=0; i<str.length(); i++){

if(!Character.isWhitespace(str.charAt(i))){

str1.append(str.charAt(i));

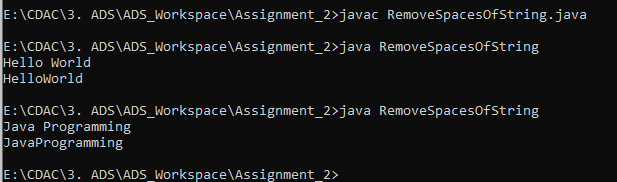
}

}

System.out.println(str1);

}

}



4. Reverse a String

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

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

/\* reverse a String \*/

import java.util.Scanner;

class ReverseString{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

String reverseStr = "";

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

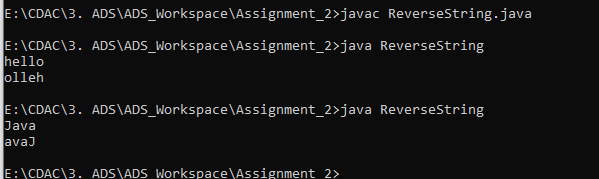
reverseStr = reverseStr + str.charAt(i);

}

System.out.println(reverseStr);

}

}



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]

/\* Reverse an array \*/

import java.util.Scanner;

class ReverseArray{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int size = sc.nextInt();

int[] arr = new int[size];

for(int i=0; i<arr.length; i++){

arr[i] = sc.nextInt();

}

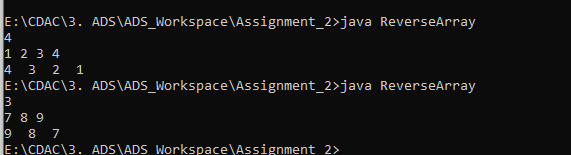
for(int i=arr.length-1; i>=0; i--){

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

}

}

}



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"

/\* Reverse words of String \*/

import java.util.Scanner;

class ReverseWordsOfString{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

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

String revWords= "";

for(int i=strArray.length - 1; i>=0; i--){

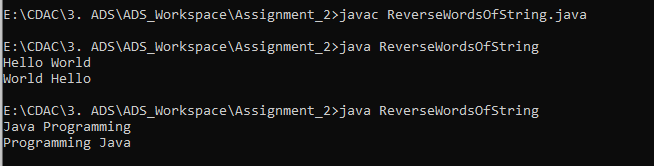
revWords = revWords + strArray[i] + " ";

}

System.out.println(revWords);

}

}



7. Reverse a Number

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

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

/\* Reverse a Number \*/

import java.util.Scanner;

class ReverseNumber{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int num = sc.nextInt();

boolean isNegative= num < 0;

if(isNegative){

num = -num;

}

int reverse = 0;

while(num >0){

int rem = num % 10;

reverse = reverse \*10 +rem;

num = num/10;

}

if(isNegative){

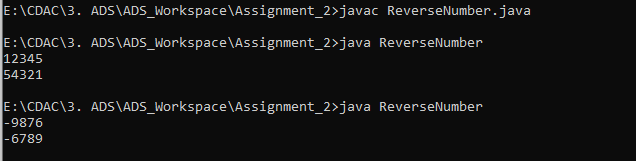
reverse = -reverse;

}

System.out.println(reverse);

}

}



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

/\* Manipulate Array based on range update queries \*/

class ArrayRange{

static int manipulateArray(int n, int[][] arr){

int[] arr1 = new int[n+1];

for(int[] element : arr){

int start = element[0];

int end = element[1];

int value = element[2];

arr1[start -1] += value;

if(end<n)

arr1[end] -= value;

}

int max = 0;

int current = 0;

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

current = current + arr1[i];

if(current > max)

max = current;

}

return max;

}

public static void main(String args[]){

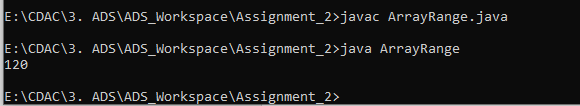
int n=4;

int[][] arr = {{1, 3, 50}, {2, 4, 70}};

System.out.println(manipulateArray(n, arr));

}

}



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:

/\* Check if the String is palindrome \*/

import java.util.Scanner;

class StringPalindrome{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

String str = sc.nextLine();

String reverse = "";

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

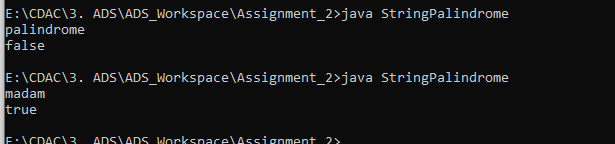
reverse = reverse + str.charAt(i);

}

System.out.println(reverse.equals(str));

}

}



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]

/\* Array Rotation to the left by d positions \*/

class ArrayRotation{

static void rotateLeft(int[] arr){

int n= arr.length;

int element = arr[0];

for(int i=0; i<n-1;i++){

arr[i] = arr[i+1];

}

arr[n-1] = element; // shift element from first position to last

}

static void rotateArray(int[] arr, int d){

if(d==0)

return;

rotateLeft(arr);

rotateArray(arr, d-1); // recursive call

}

public static void main(String args[]){

int[] arr = {1, 2, 3, 4, 5};

int d = 2;

rotateArray(arr, d);

System.out.print("[");

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i]);

if (i < arr.length - 1) {

System.out.print(", ");

}

}

System.out.println("]");

}

}

