**C-DAC Mumbai**

**Date 26/09/2024**

**Subject: Algorithm and Data Structure**

**Assignment 2**

**Rohit Owal**

**CDAC Mumbai**

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:**

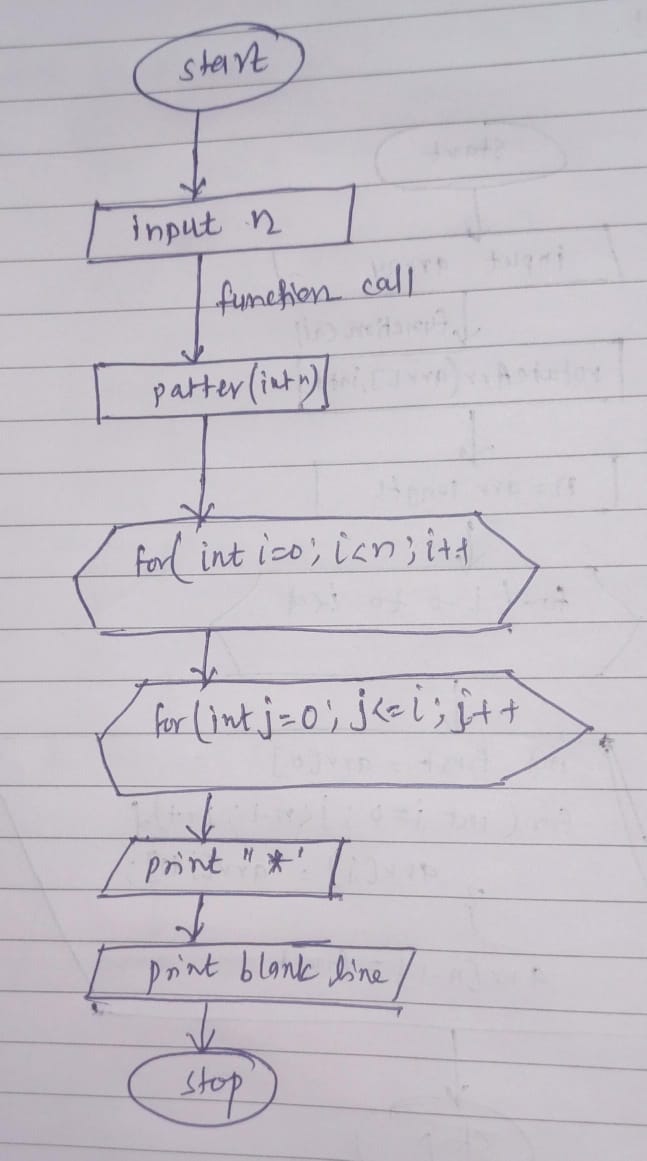
**\***

**\*\***

**\*\*\***

**\*\*\*\***

**\*\*\*\*\***



class Q1Pattern{

static void pattern(int n){

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

for(int j=0;j<=i;j++){ //time complexity = O(n^2)

System.out.print("\* "); //space complexity =O(1)

}

System.out.println();

}

}

public static void main(String args[]){

int n1 = 5;

int n2 = 3;

System.out.println("When n = "+n1);

pattern(n1);

System.out.println();

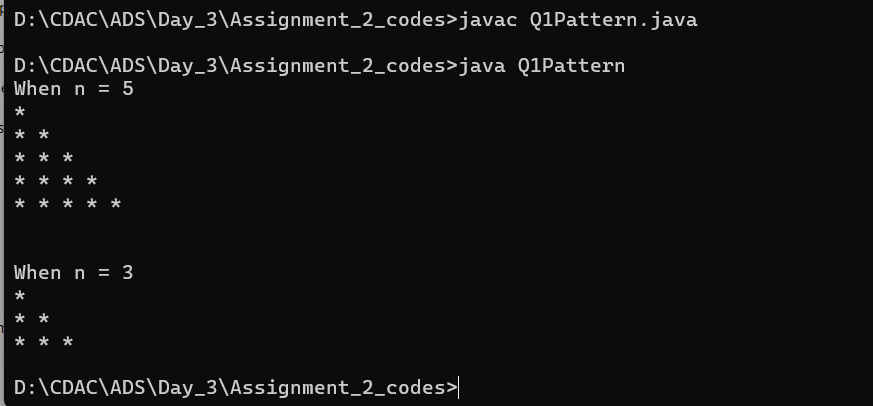
System.out.println();

System.out.println("When n = "+n2);

pattern(n2);

}

}



**2. Remove Array Duplicates**

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

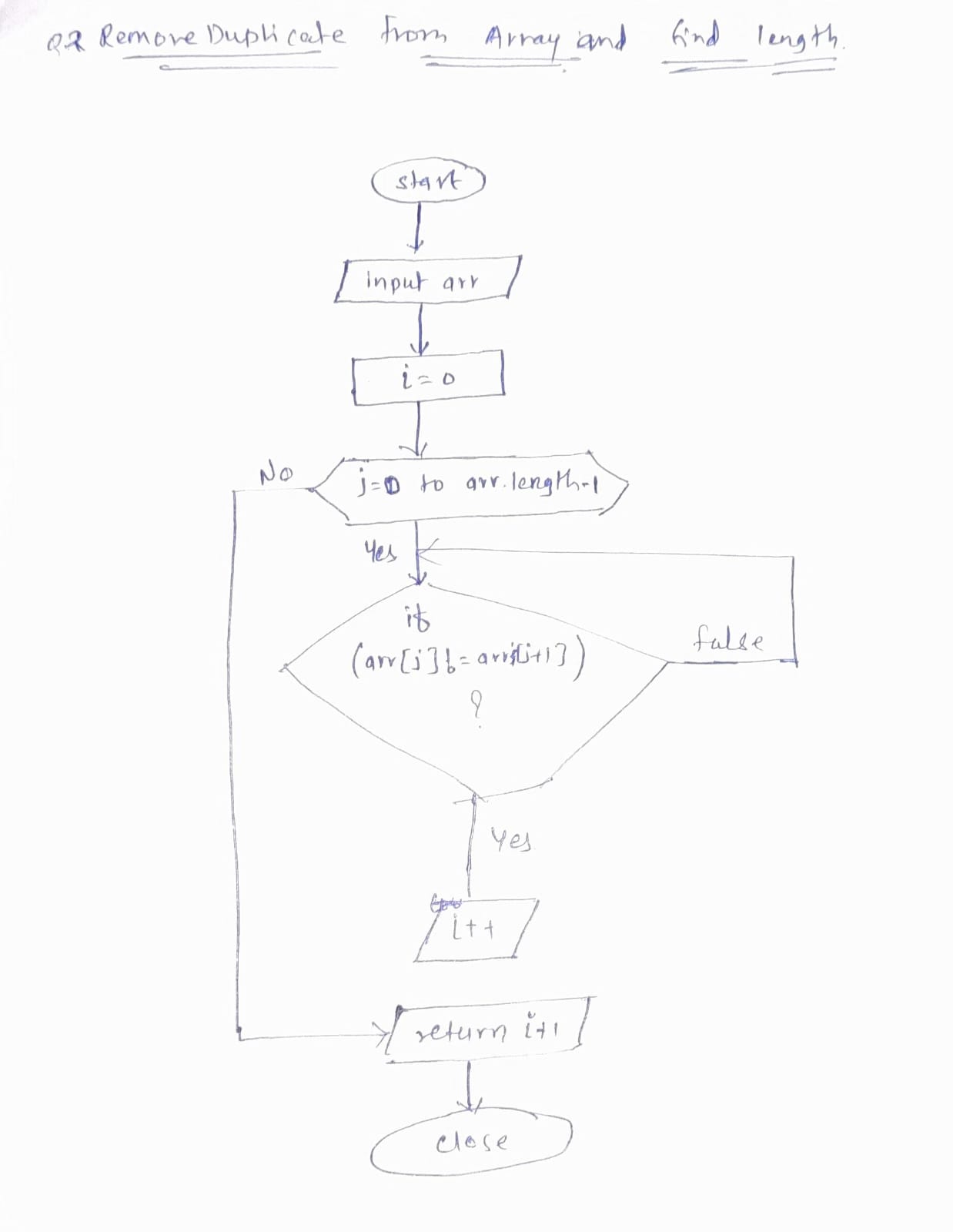
**Test Cases:**

**Input: arr = [1, 1, 2]**

**Output: 2**

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

**Output: 4**



class Q2RemoveDuplicate{

static int lengthOfDeupliacte(int [] arr ){

int i=0;

for(int j=1;j<arr.length;j++){ //time complexity = O(n)

if(arr[i]!=arr[j]){ //space complexity = O(1)

i++;

arr[i]=arr[j];

}

}

return i+1;

}

public static void main(String args[]){

int arr1 [] = {1, 1, 2};

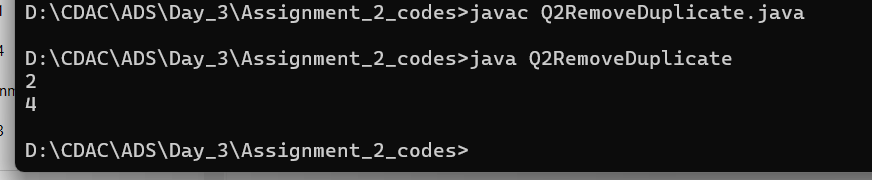
int arr2 [] = {0, 0, 1, 1, 2, 2, 3, 3};

System.out.println(lengthOfDeupliacte(arr1));

System.out.println(lengthOfDeupliacte(arr2));

}

}



**3. Remove White Spaces from String**

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

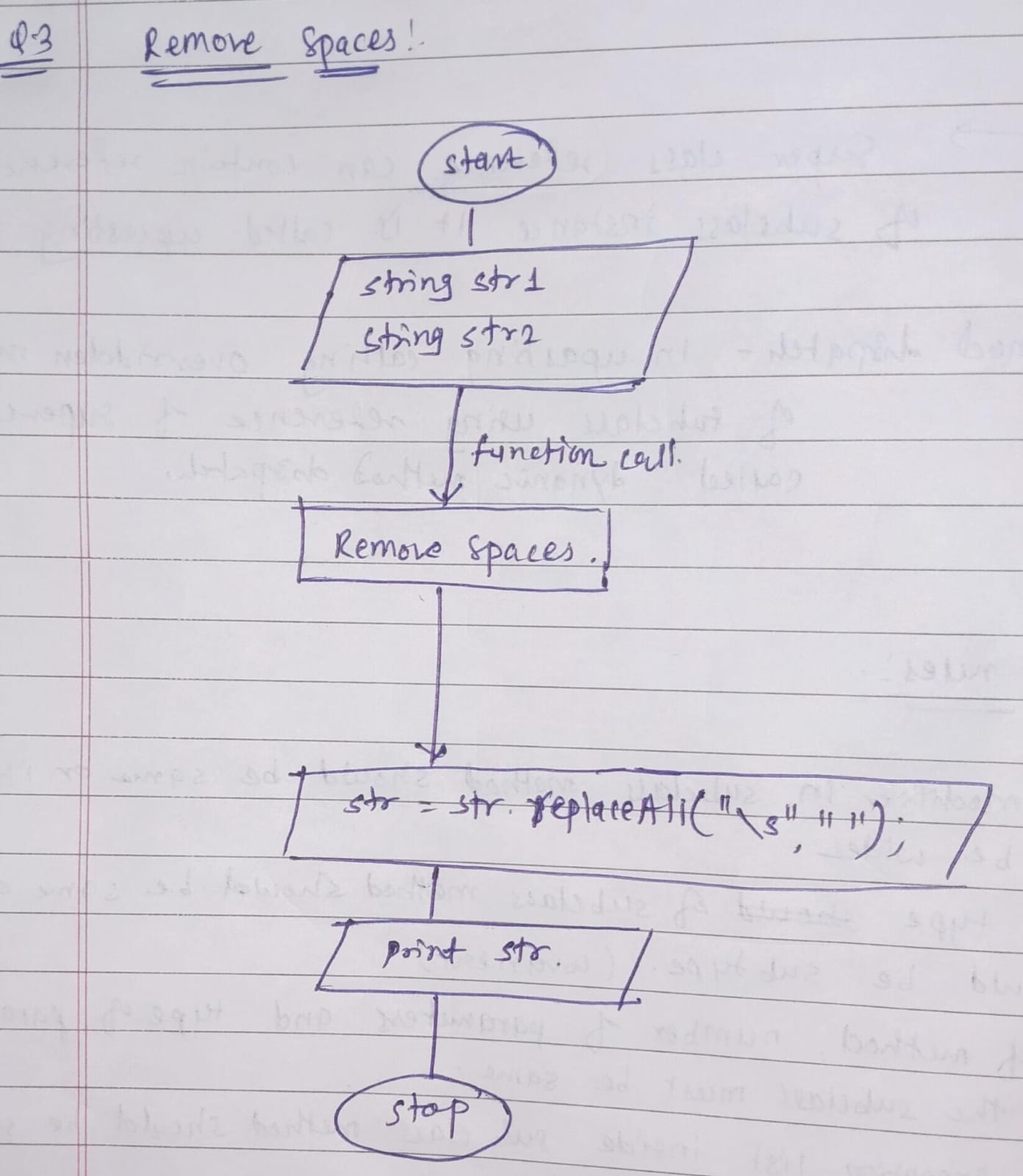
**Test Cases:**

**Input: "Hello World"**

**Output: "HelloWorld"**

**Input: " Java Programming "**

**Output: "JavaProgramming"**



class Q3RemoveSpaces{

static String removeSpace(String str){ //time complexity = O(n)

str = str.replaceAll("\\s",""); //space complexity = O(n)

return str;

}

public static void main(String args [] ){

String str1 = "Hello World";

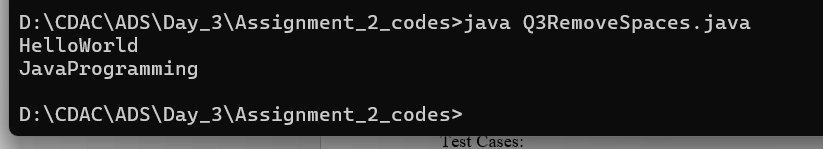
String str2 = " Java Programming ";

System.out.println(removeSpace(str1));

System.out.println(removeSpace(str2));

}

}



**4. Reverse a String**

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

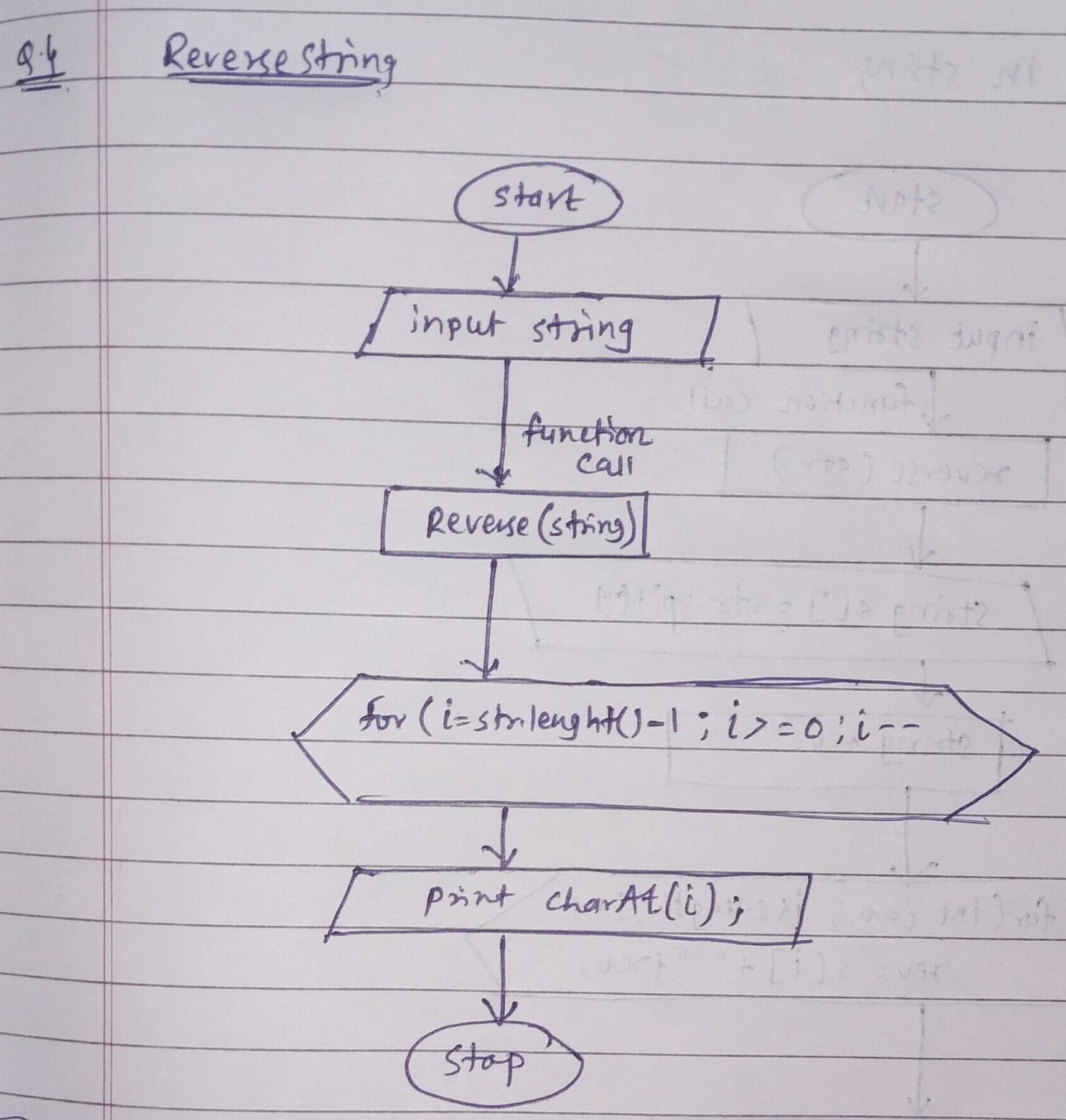
**Test Cases:**

**Input: "hello"**

**Output: "olleh"**

**Input: "Java"**

**Output: "avaJ"**



class Q4ReverseString{

static void reverse(String str){

for(int i=str.length()-1;i>=0;i--){ //time complexity = O(n)

System.out.print(str.charAt(i)); //space complexity = O(1)

}

}

public static void main(String args[]){

String str1 = "hello";

String str2 = "Java";

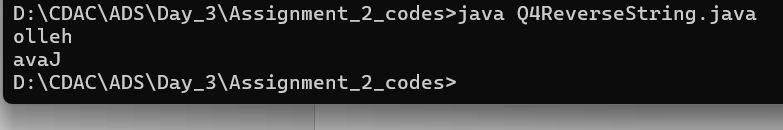
reverse(str1);

System.out.println();

reverse(str2);

}

}



**5. Reverse Array in Place**

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

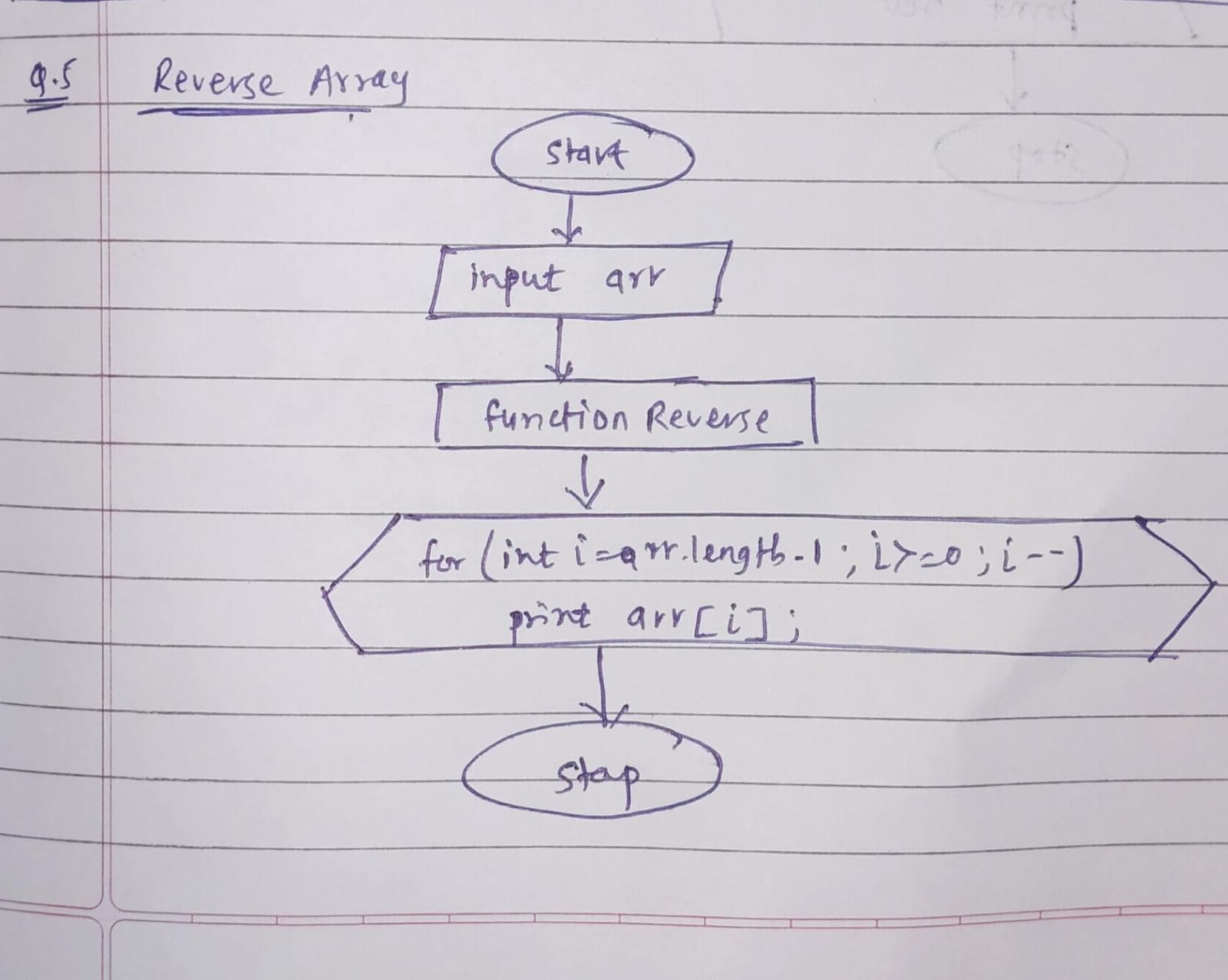
**Test Cases:**

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

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

**Input: arr = [7, 8, 9]**

**Output: [9, 8, 7]**



class Q5ReverseArray{

static void reverse(int arr [] ){

for(int i = arr.length-1;i>=0;i--){ //time complexity = O(n)

System.out.print(arr[i]+" "); //space complexity = O(1)

}

}

public static void main(String args []){

int arr1 [] = {1,2,3,4};

int arr2 [] = {7,8,9};

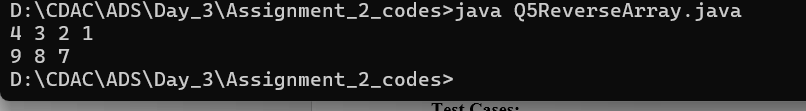
reverse(arr1);

System.out.println();

reverse(arr2);

}

}



**6. Reverse Words in a String**

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

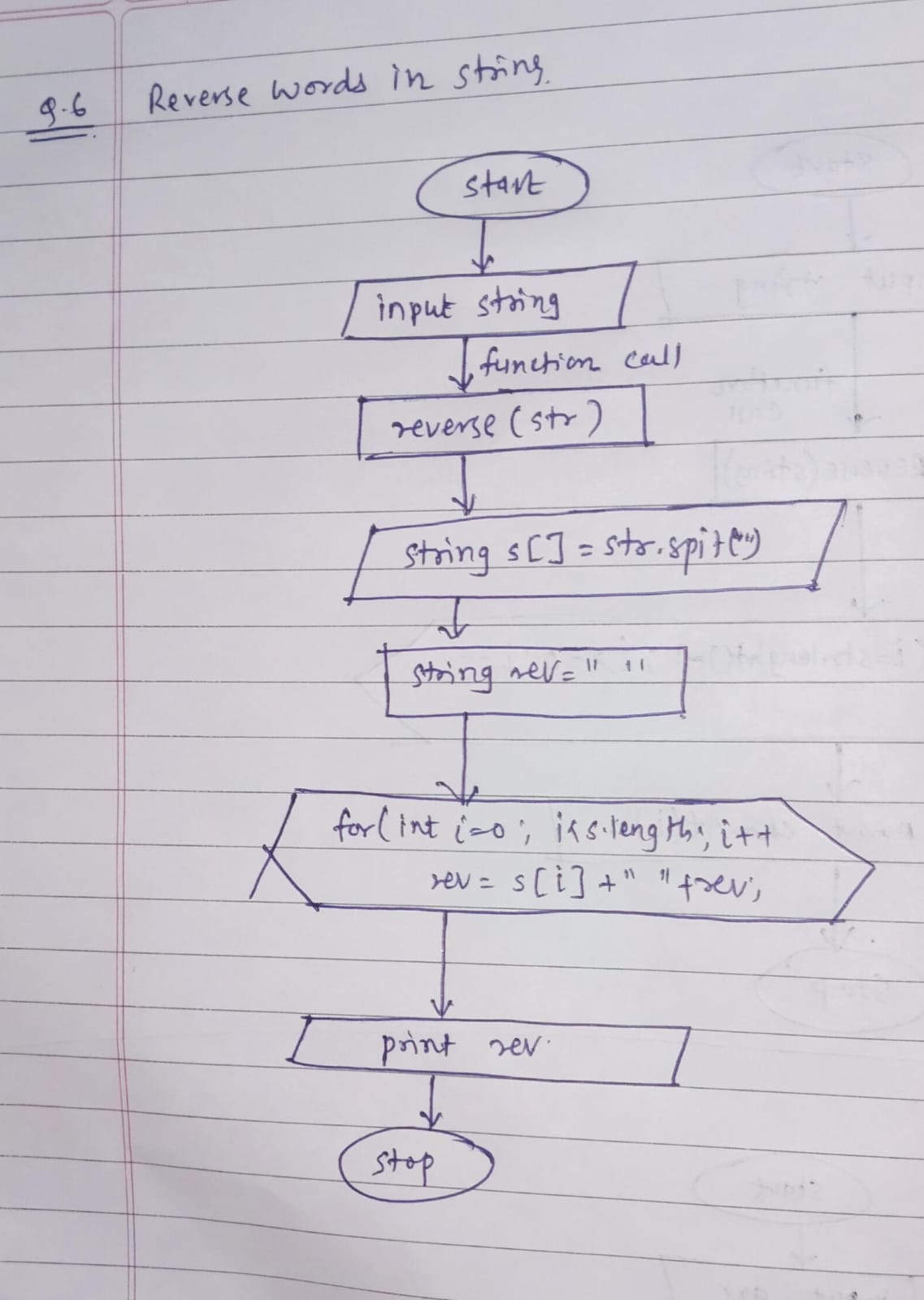
**Test Cases:**

**Input: "Hello World"**

**Output: "World Hello"**

**Input: "Java Programming"**

**Output: "Programming Java"**



class Q6Reverse {

static void reverse(String str){

//Breaking the sentence into words

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

String rev = " ";

for(int i=0;i<s.length;i++){ //tc=O(n) sc=O(n)

rev = s[i]+" "+rev;

}

System.out.println(rev);

}

public static void main(String[] args) {

String str1 = "Hello World";

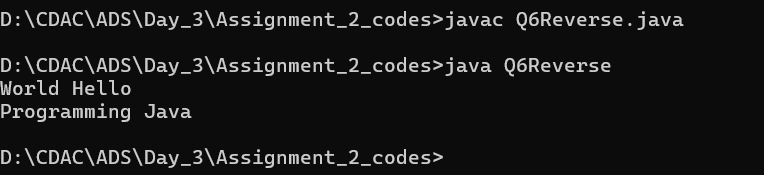
String str2 = "Java Programming";

reverse(str1);

reverse(str2);

}

}



**7. Reverse a Number**

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

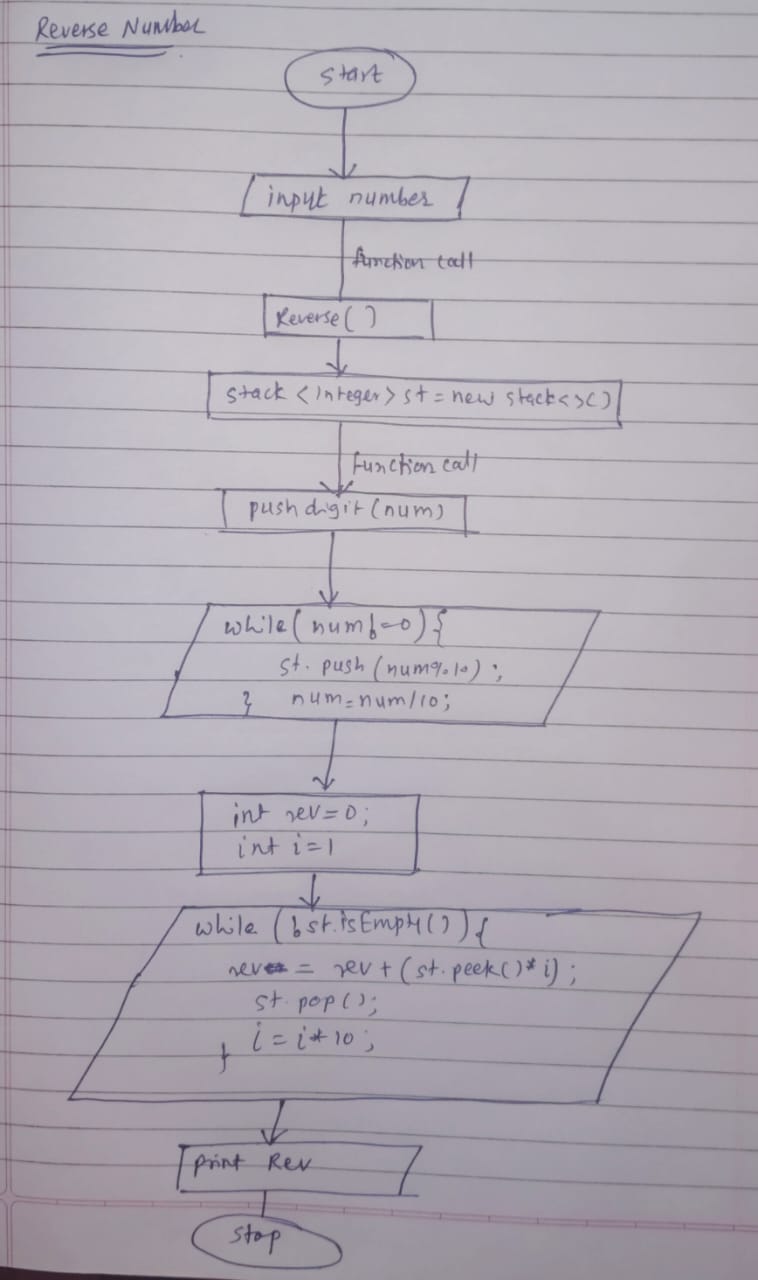
**Test Cases:**

**Input: 12345**

**Output: 54321**

**Input: -9876**

**Output: -6789**



import java.util.\*;

class Q7ReverseNumber{

static Stack<Integer> st = new Stack<>();

static void pushDigit(int num){ //time complexity = O(n)

while(num!=0){ //space complexity=O(n)

st.push(num%10);

num = num/10;

}

}

static int reverse(int num){

pushDigit(num);

int reverse = 0;

int i=1;

while (!st.isEmpty())

{

reverse = reverse + (st.peek() \* i);

st.pop();

i = i \* 10;

}

return reverse;

}

public static void main(String args []){

int num1 = 12345;

int num2 = -9876;

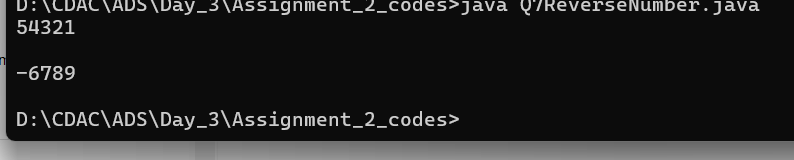
System.out.println(reverse(num1));

System.out.println();

System.out.println(reverse(num2));

}

}



**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**

class Q8ArrayManipulation{

static long manipulation(int n, int [] [] queries1){

long res [] = new long[n+2]; //space complexity = O(n);

for(int i=0;i<queries1.length;i++){ //time complexity = O(n);

int a = queries1[i][0];

int b = queries1[i][1];

int k = queries1[i][2];

res[a] = res[a]+k;

res[b+1] = res[b+1]-k;

}

long max = 0;

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

res[i]=res[i]+res[i-1];

max = Math.max(res[i],max);

}

return max;

}

public static void main(String args[]){

int n1 = 5; //size of array

int[][] queries1 = {{1, 2, 100}, {2, 5, 100}, {3, 4, 100}};

int n2 = 4;

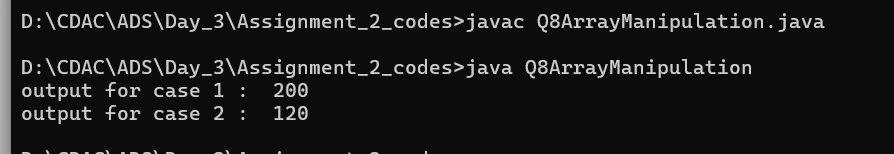
int[][] queries2 = {{1, 3, 50}, {2, 4, 70}};

System.out.println("output for case 1 : "+manipulation(n1,queries1));

System.out.println("output for case 2 : "+manipulation(n2,queries2));

}

}



**9. String Palindrome**

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

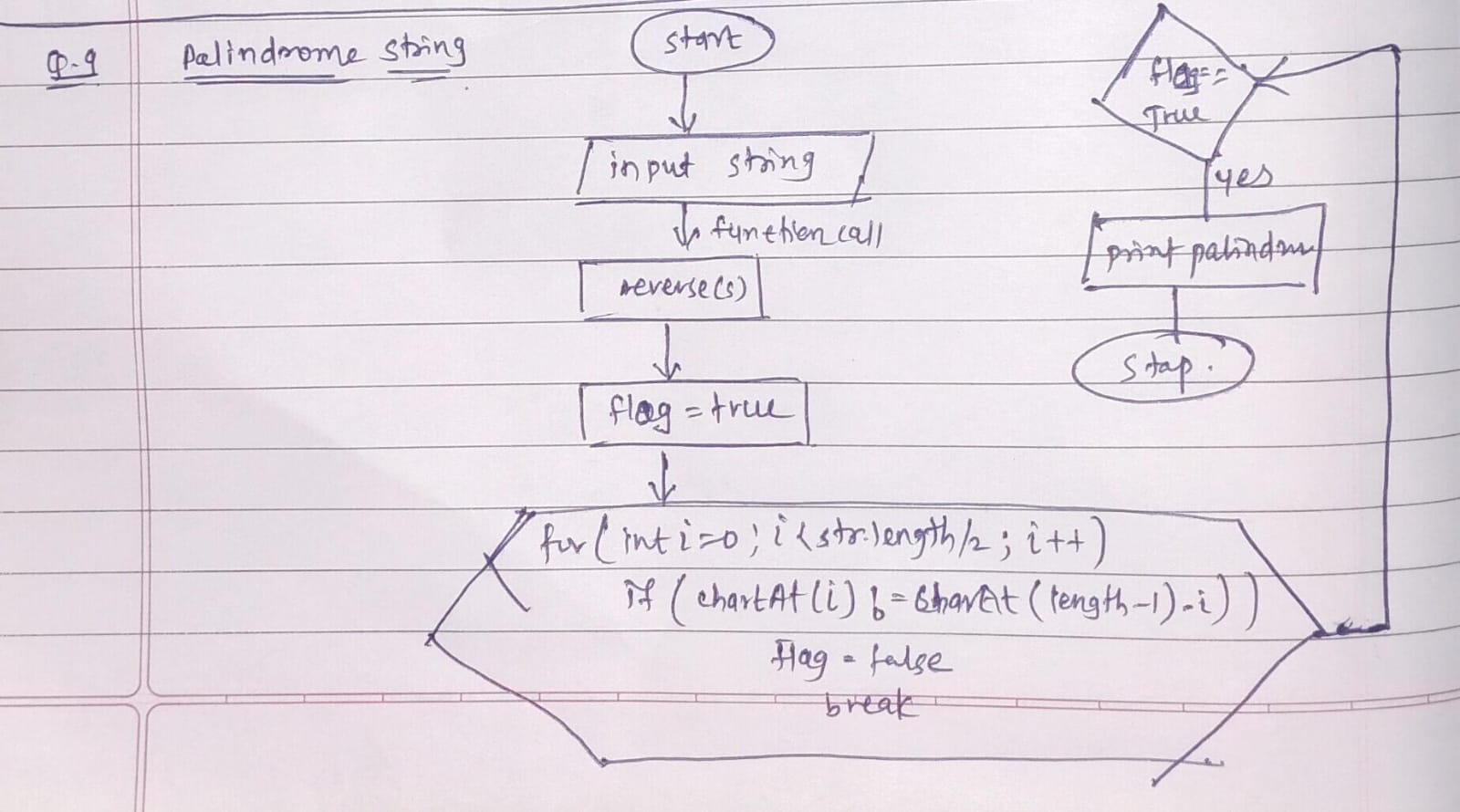
**Test Cases:**

**Input: "madam"**

**Output: true**

**Input: "hello"**

**Output: false**



class Q9PalindromeString{

static void reverse(String str){

boolean flag = true;

for(int i=0;i<str.length()/2;i++){ //time complexity = O(n)

if(str.charAt(i)!= str.charAt(str.length()-i-1)) //space complexity = O(1)

flag = false;

break;

}

if(flag==true)

System.out.println("True");

else

System.out.println("False");

}

public static void main(String args[]){

String str1 = "madam";

String str2 = "hello";

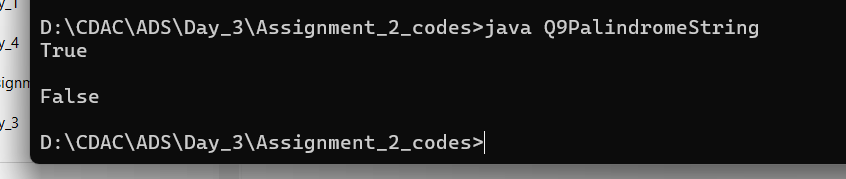
reverse(str1);

System.out.println();

reverse(str2);

}

}



**10. Array Left Rotation**

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

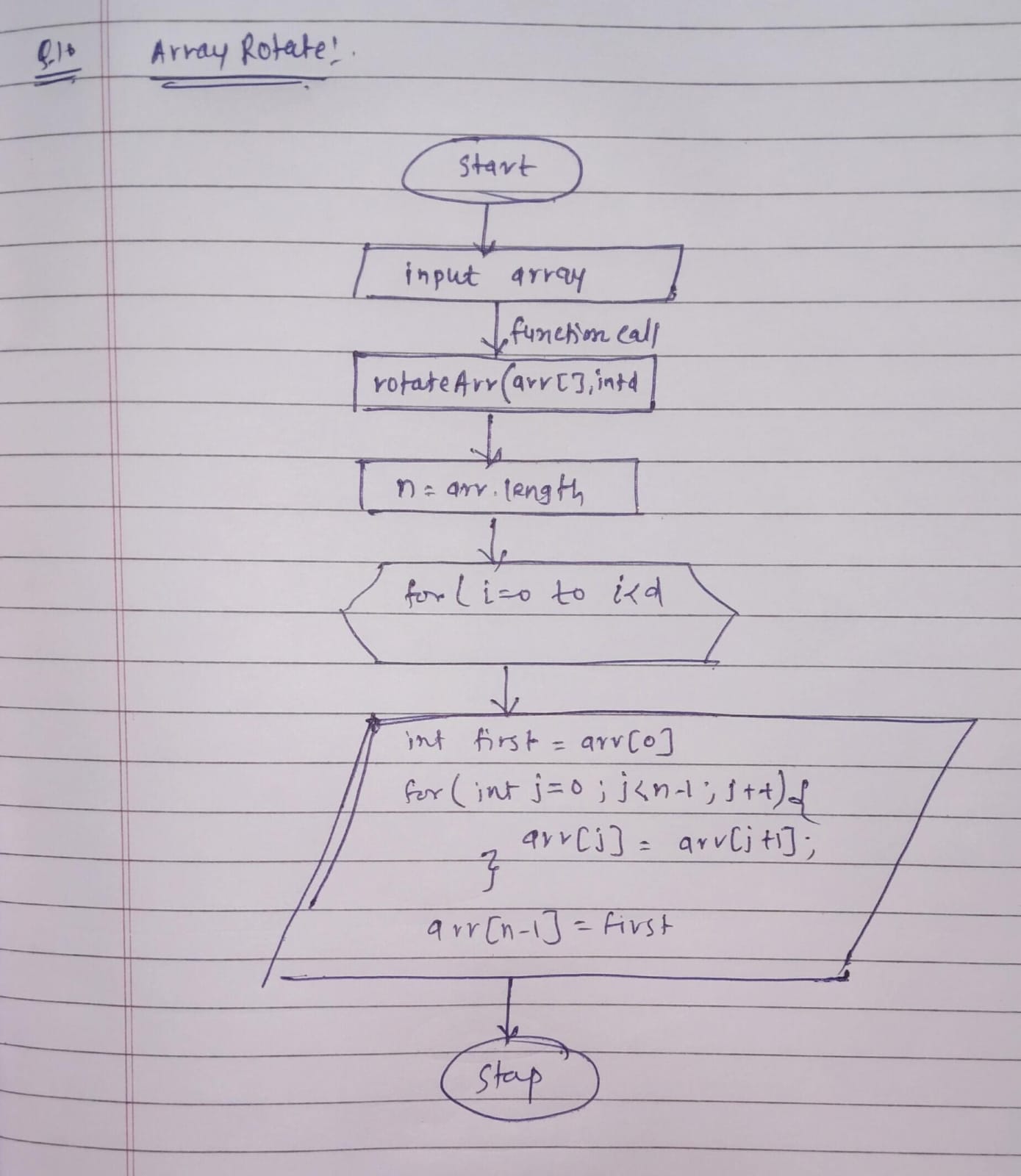
**Test Cases:**

**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]**



class Q10ROtateArray{

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

int n = arr.length;

for(int i=0;i<d;i++){ //TC= O(n\*d)

int first = arr[0]; //SC=O(1)

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

arr[j] = arr[j+1];

}

arr[n-1] = first;

}

}

public static void main(String args[]){

int [] arr1 = {1, 2, 3, 4, 5};

int d1 = 2;

rotateArr(arr1,d1);

for(int ele : arr1)

System.out.print(ele+" ");

System.err.println();

int [] arr2 = {10,20,30,40};

int d2 = 1;

rotateArr(arr2,d2);

for(int ele : arr2)

System.out.print(ele+" ");

}

}

