**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:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

**Program:**

import java.util.Scanner;

public class Ass2prg1{

public static void main(String[]args){

int rows;

System.out.println("enter rows");

Scanner sc = new Scanner(System.in);

rows =sc.nextInt();

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

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

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

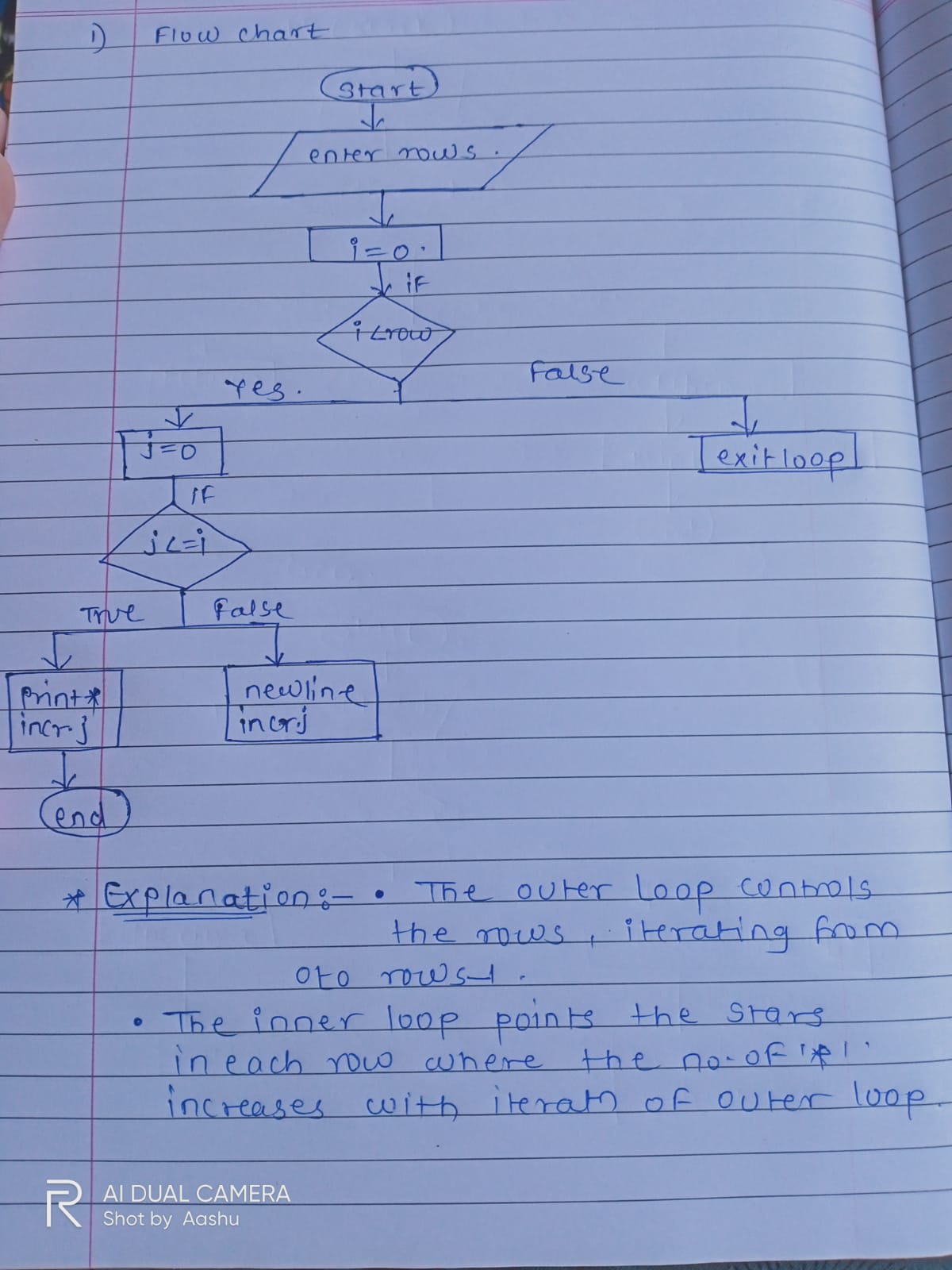
}

System.out.println();

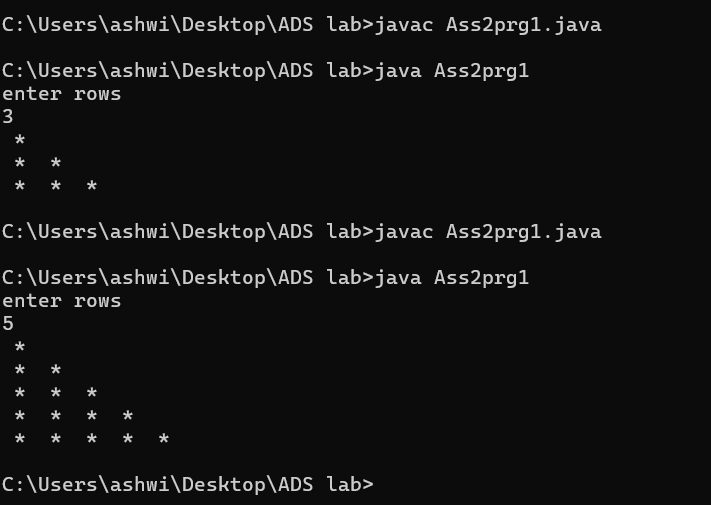
}

sc.close();

}

}

**Output:**



**Complexity:**

The inner loop runs i+1 times for each iteration of the outer loop, where i is the current row.

Thus, the total number of iterations is:

T(n)=1+2+3+⋯+n = n(n+1)/2

So, the time complexity is O(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

**Program:**

import java.util.Scanner;

public class Ass2prg2 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

int[] arr = new int[size];

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

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

arr[i] = scanner.nextInt();

}

int newLength = 1;

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

if (arr[i] != arr[i - 1]) {

arr[newLength] = arr[i];

newLength++;

}

}

System.out.println("New length: " + newLength);

System.out.print("Array without duplicates: ");

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

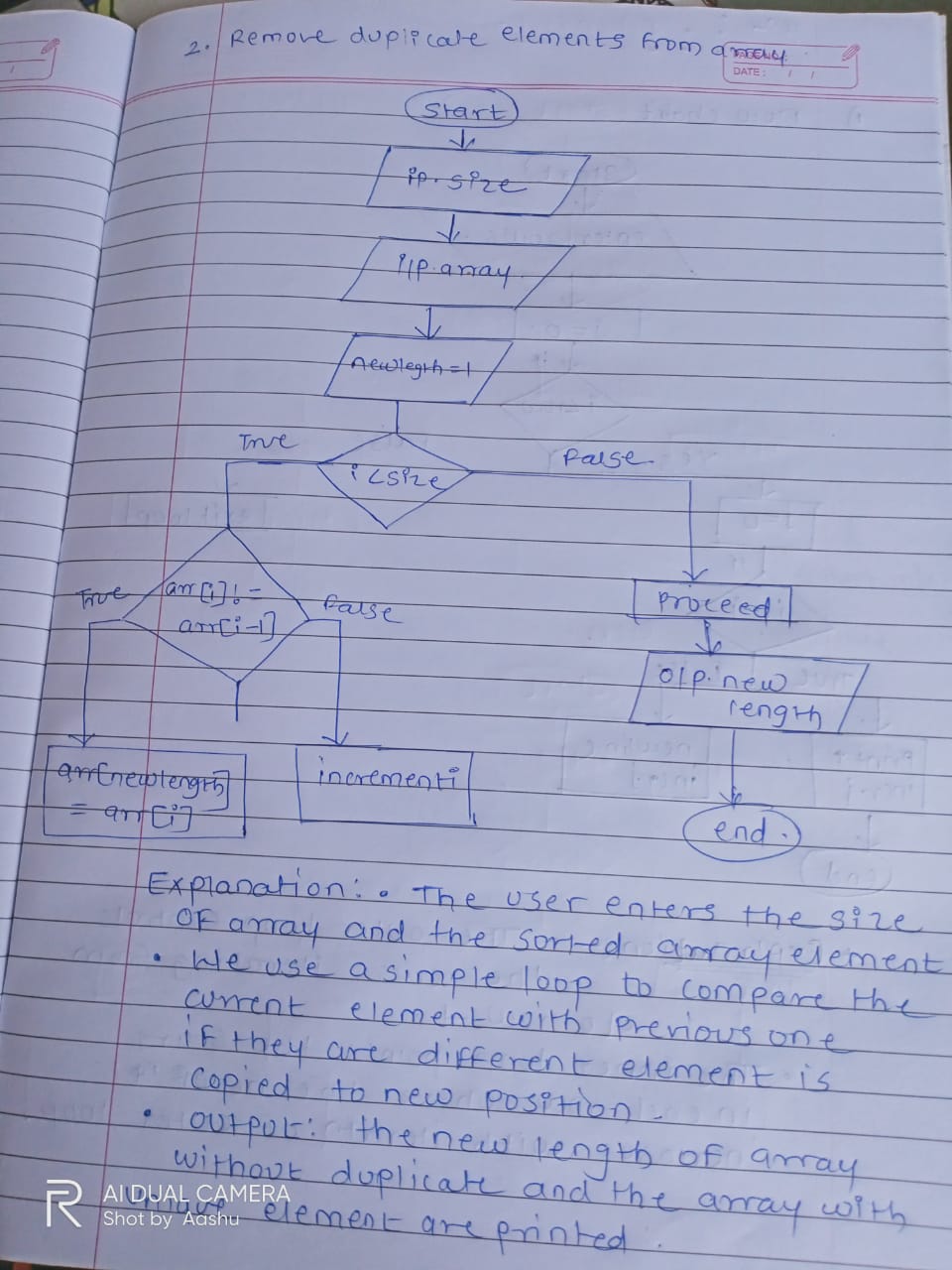
System.out.print(arr[i] + " ");

}

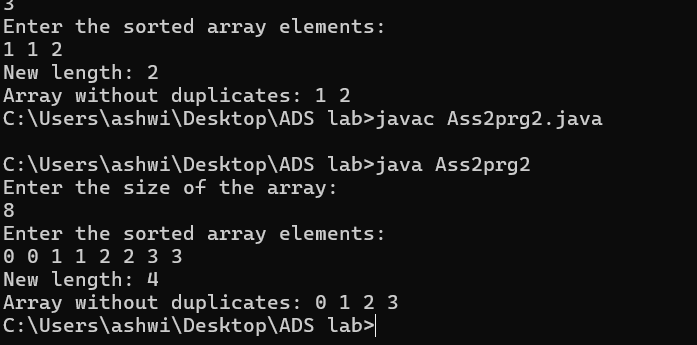
scanner.close();

}

}



**Output:**



**Complexity:**

**Time Complexity:**

O(n), where n is the size of the array we only traverse the array once

**Space Complexity:**

O(1), as we are modifying the array in place without extra space.

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"

**Program:**

public class Ass2prg3{

public static void main(String []args){

String Str1 = "Hello World";

String Str2 = " Java Programming ";

Str1= Str1.replaceAll("\\s", "");

System.out.println(Str1);

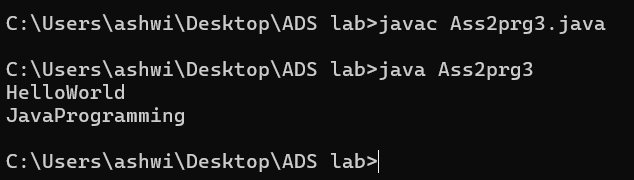
Str2 = Str2.replaceAll("\\s", "");

System.out.println(Str2);

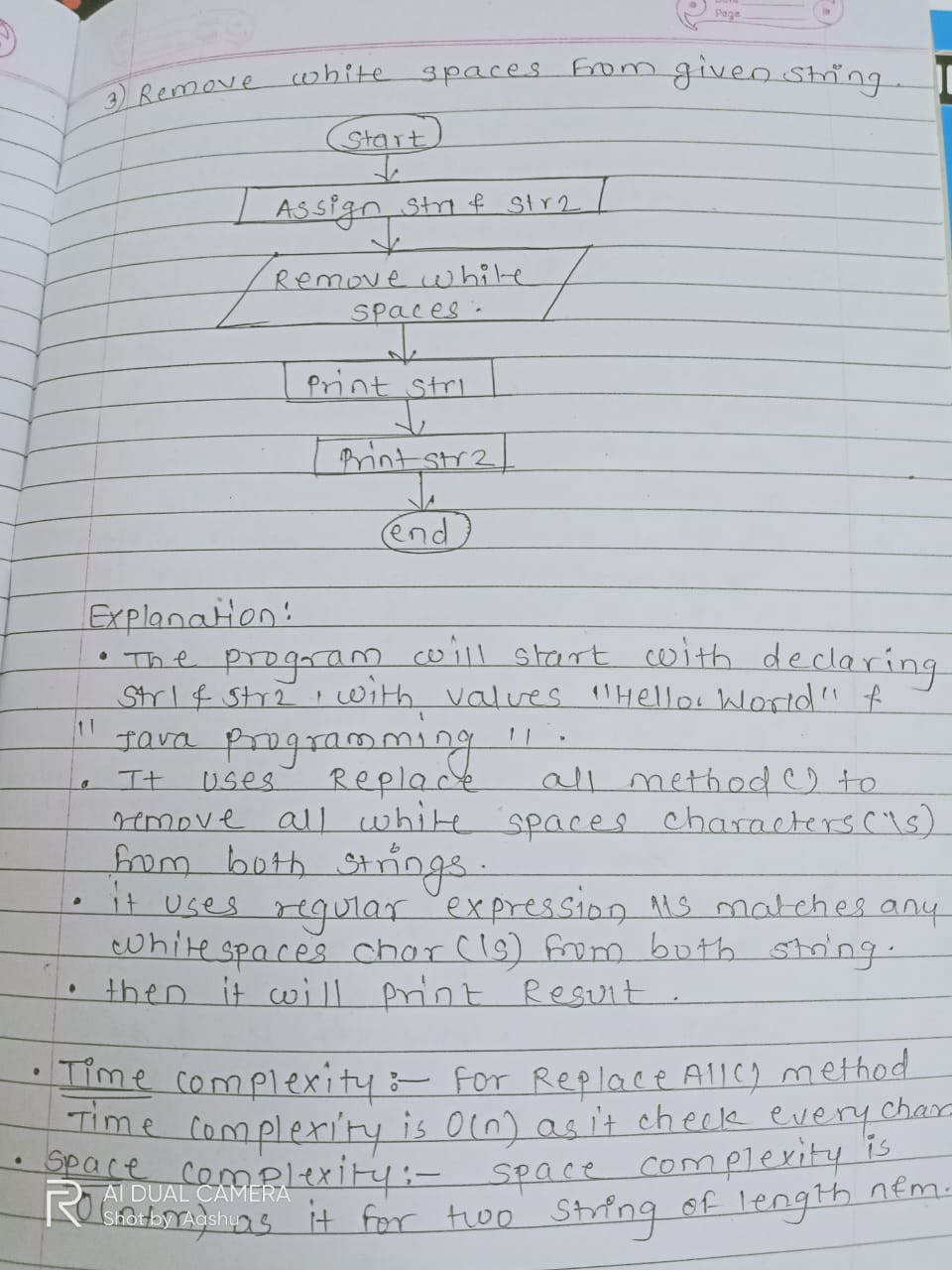
}

}

**Output:**



**Flowchart:**



4. Reverse a String

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

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

**Program:**

import java.util.Scanner;

public class Ass2prg4{

public static void main (String[] args){

String str;

String rev = "";

System.out.println("enter the string ");

Scanner sc = new Scanner (System.in);

str = sc.nextLine();

int length = str.length();

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

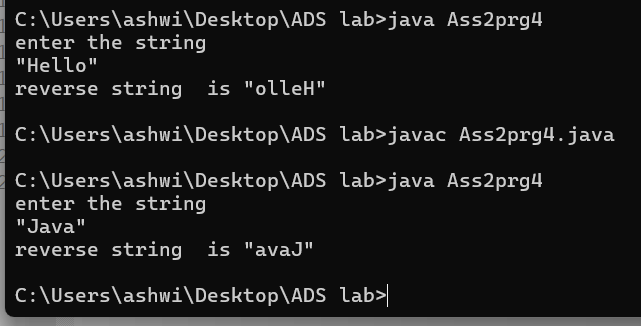
rev = rev + str.charAt(i);

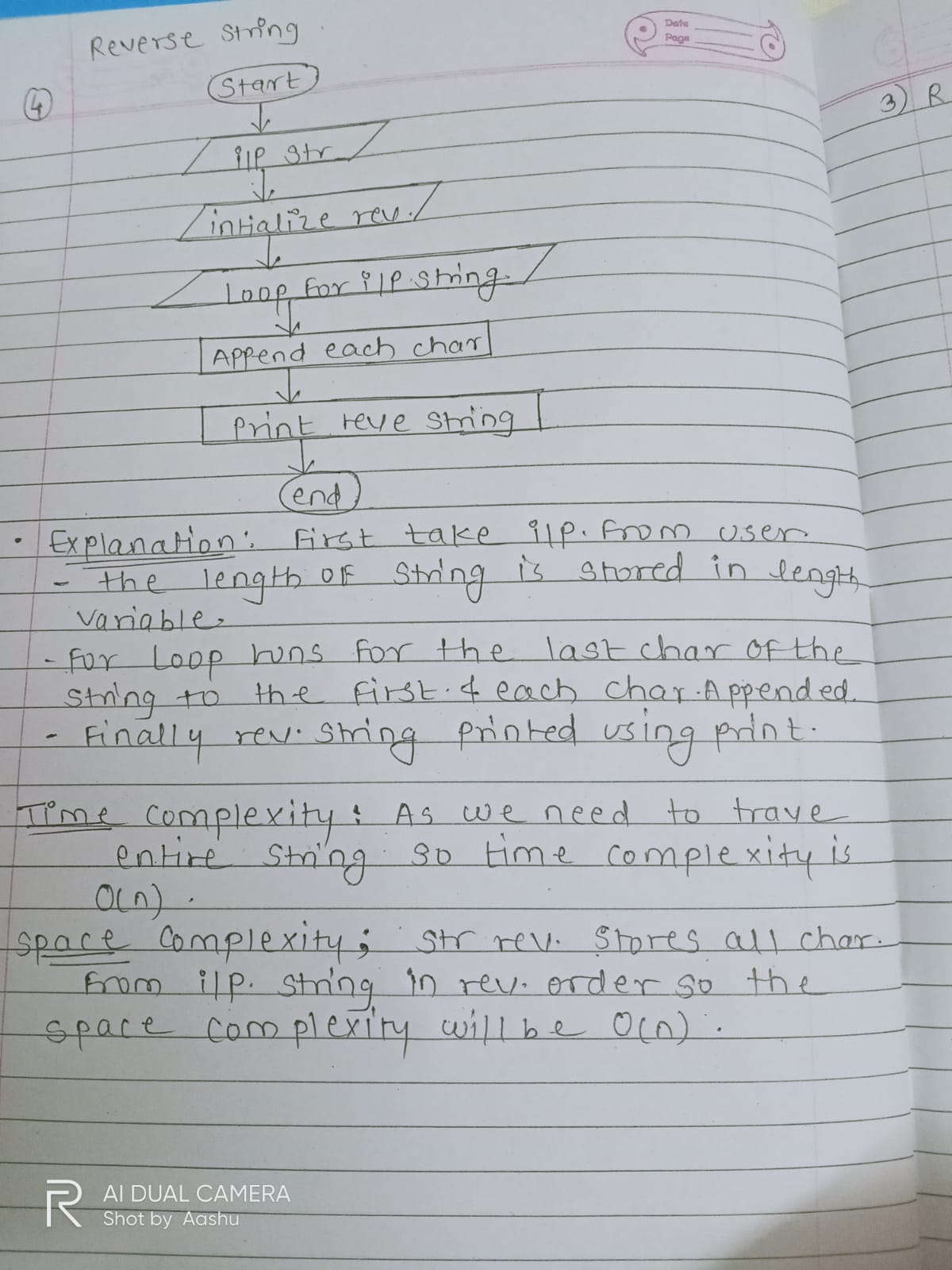
}

System.out.println("reverse string is " + rev);

}

**Output:**





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]4

**Program:**

import java.util.Scanner;

public class Ass2prg5{

public static void main(String [] args){

Scanner sc = new Scanner(System.in);

System.out.println("enter length of array");

int length= sc.nextInt();

int arr[] = new int[length];

System.out.println("enter array elements");

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

arr[i] = sc.nextInt();

}

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

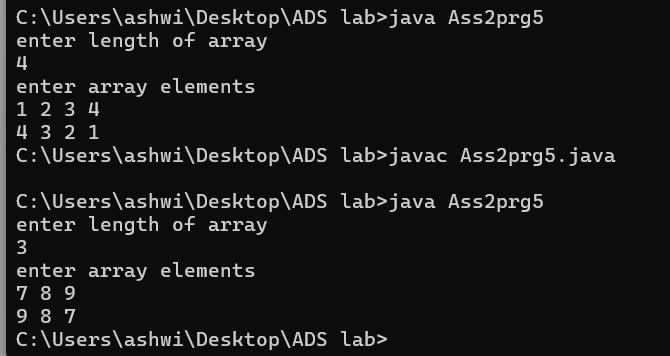
System.out.print(arr[i]+ " ");

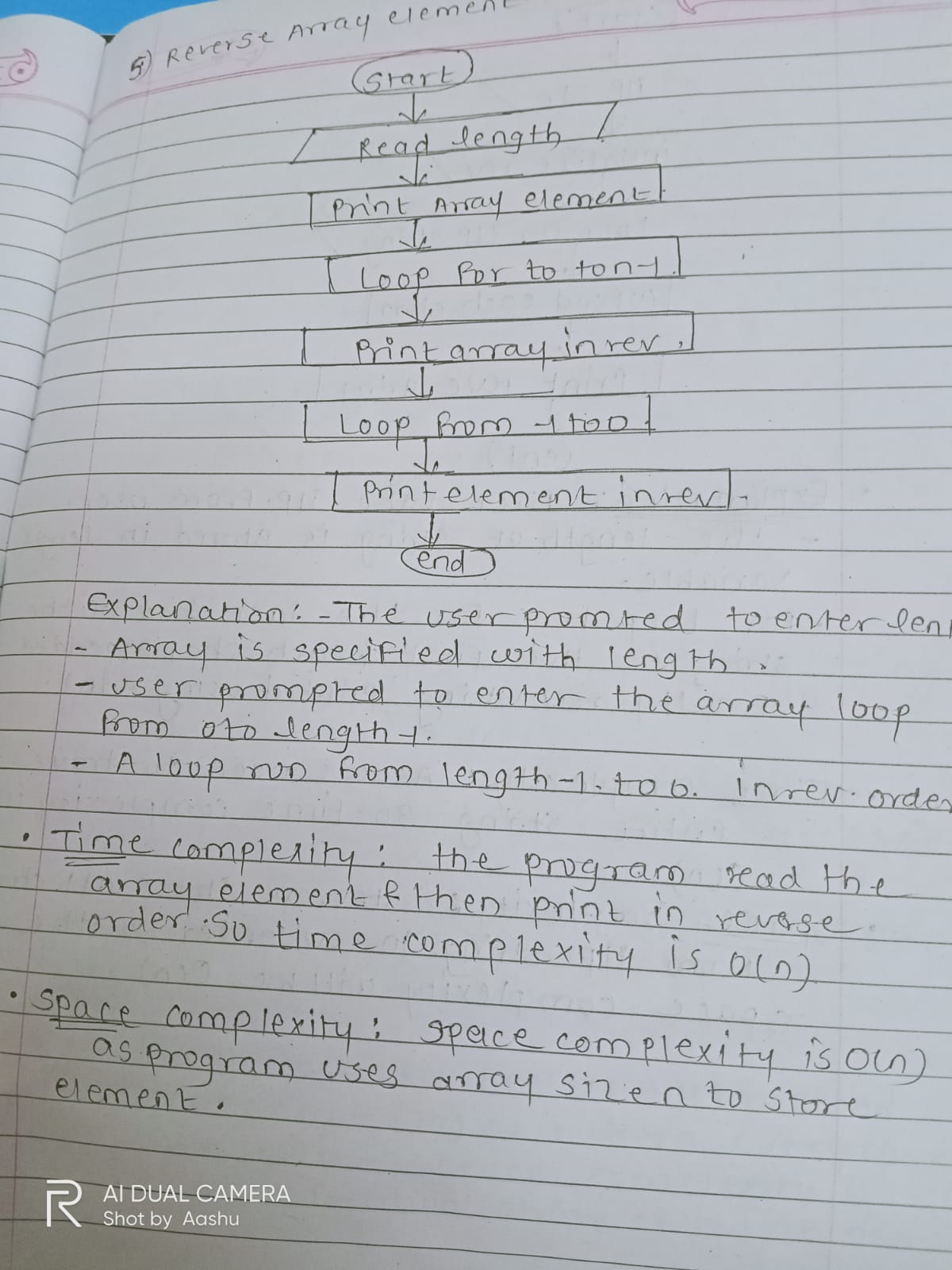
}

}

}

**Output:**





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"

**Program:**

import java.util.Scanner;

public class Ass2prg6 {

public static void main(String[] args) {

String str;

System.out.println("Enter the sentence: ");

Scanner sc = new Scanner(System.in);

str = sc.nextLine();

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

System.out.println("\nOriginal sentence:");

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

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

}

System.out.println("\n\nReversed sentence:");

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

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

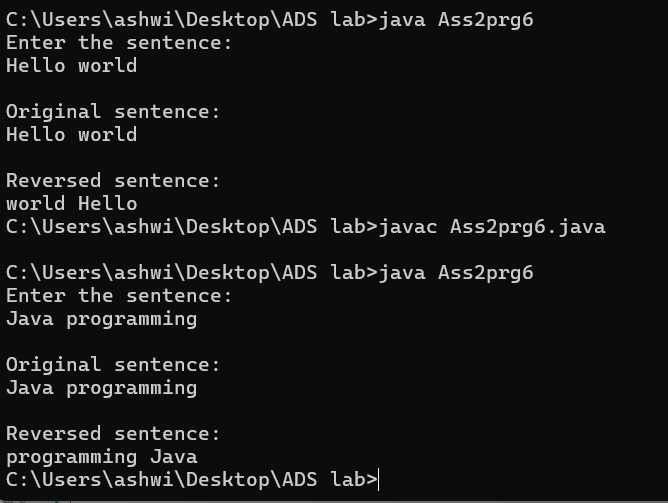
}

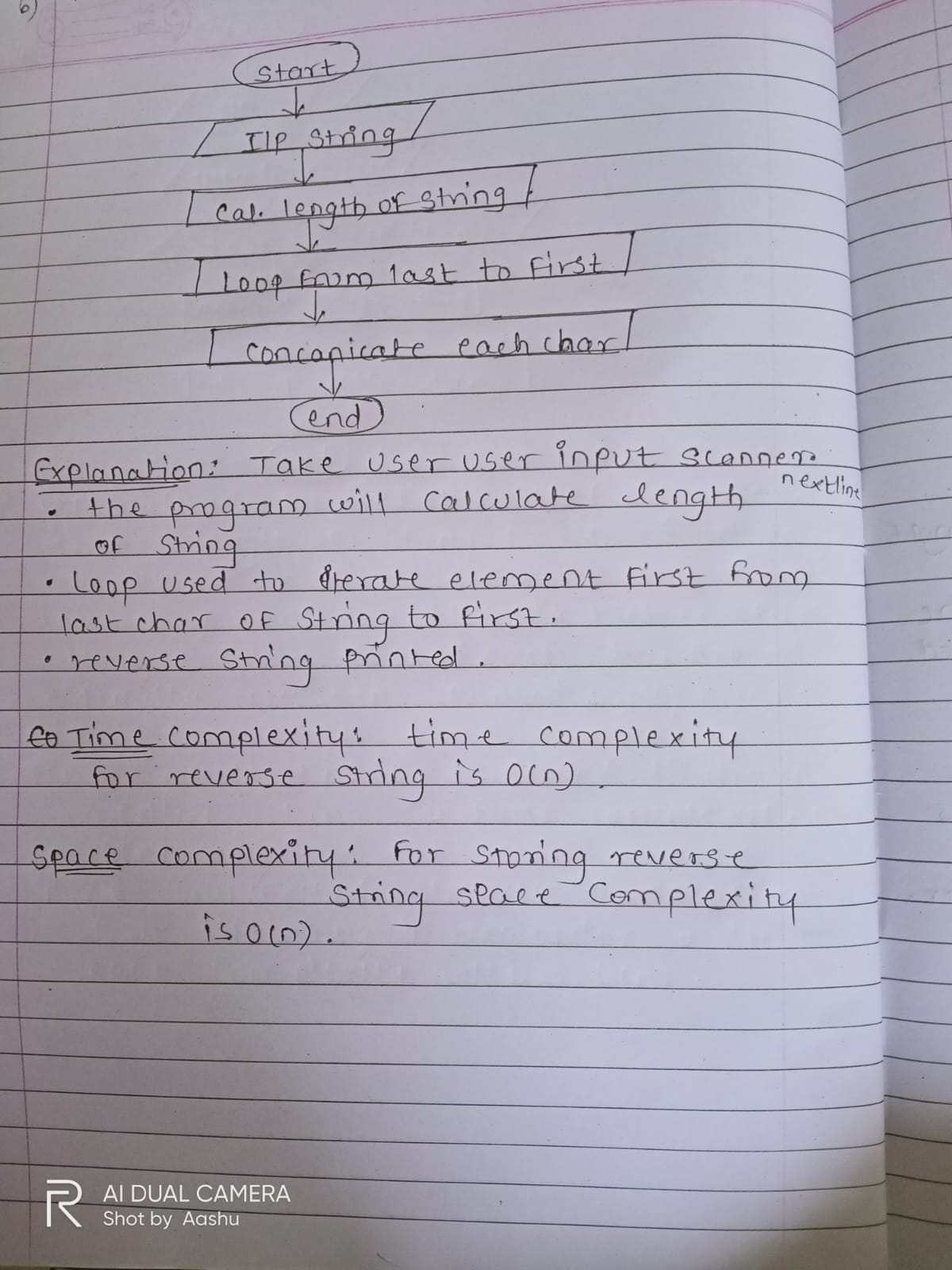
sc.close();

}

}

**Output:**





7. Reverse a Number

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

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

**Program:**

import java.util.Scanner;

public class Ass2prg7 {

public static void main(String[] args) {

int n ,r = 0 ;

System.out.print("Enter the number: ");

Scanner sc = new Scanner(System.in);

n = sc.nextInt();

if(n <0){

System.out.print("-");

n = -n;

}

while(n!=0)

{

int digit = n%10;

r = r\*10+ digit;

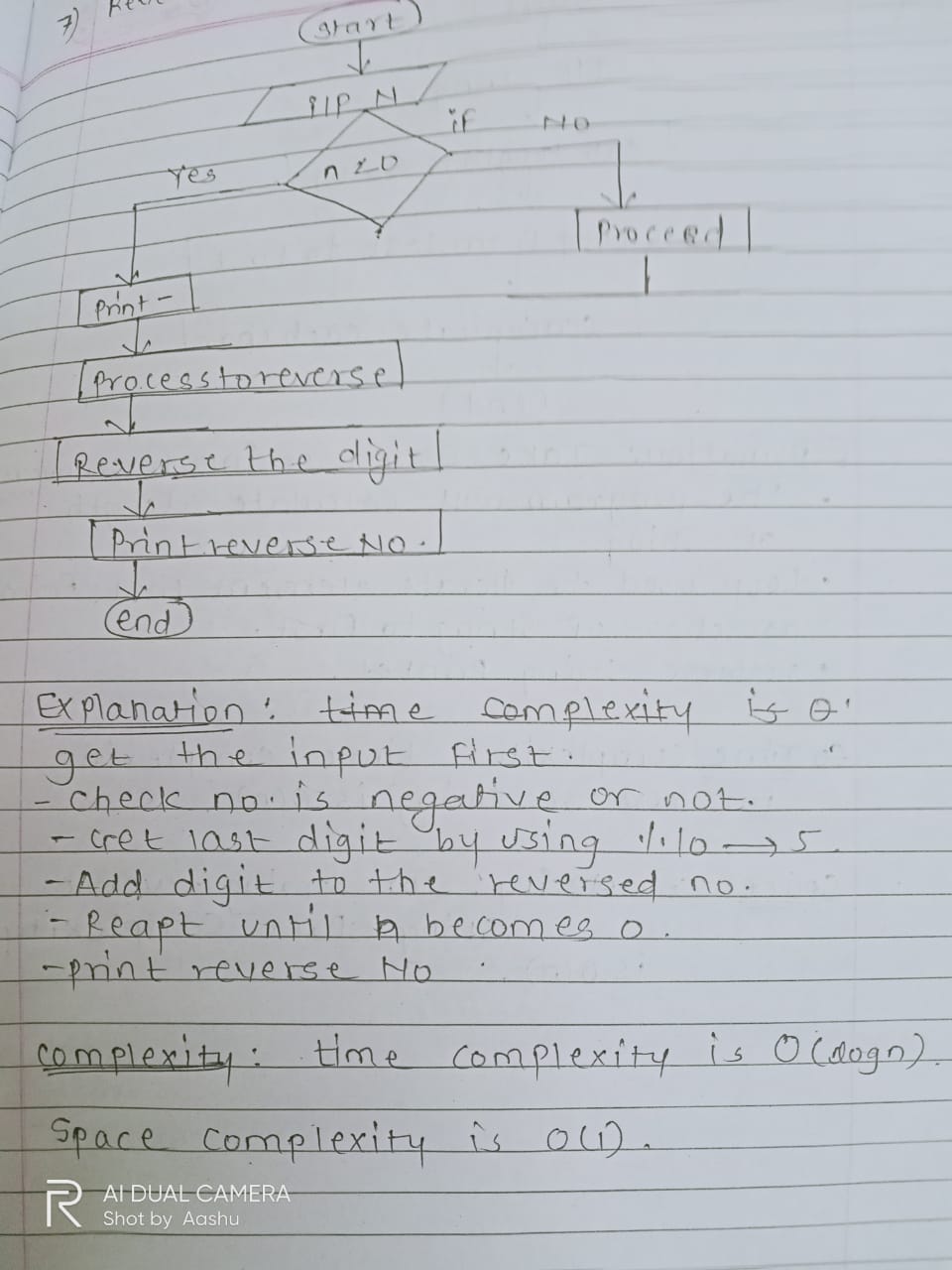
n = n/10;

}

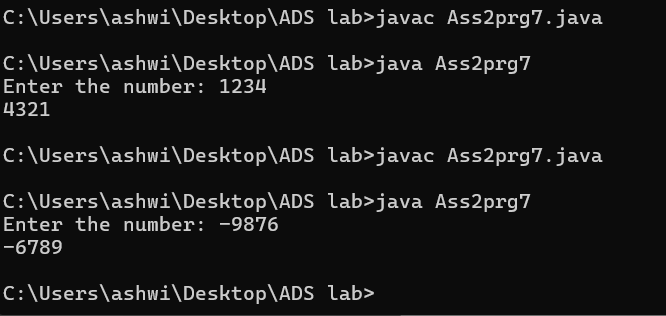
System.out.println(r);

}

}



**Output:**



8. Array Manipulation

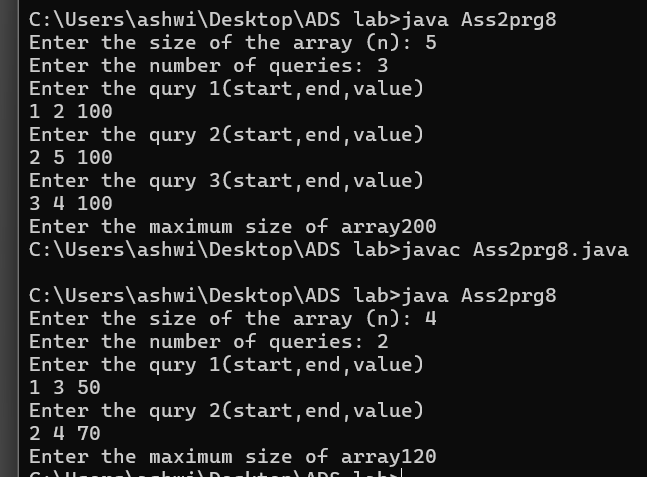
Problem: Perform a series of operations to manipulate an array based on range update queries. Each query adds a value to a range of indices.

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

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:

**Program:**

import java.util.\*;

public class Ass2prg9{

public static void main(String []args){

Scanner sc = new Scanner(System.in);

System.out.println("enter string");

String str = sc.next();

String rev= "";

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

rev = str.charAt(i) + rev;

}

if(str.equalsIgnoreCase(rev)){

System.out.println("true");

}else{

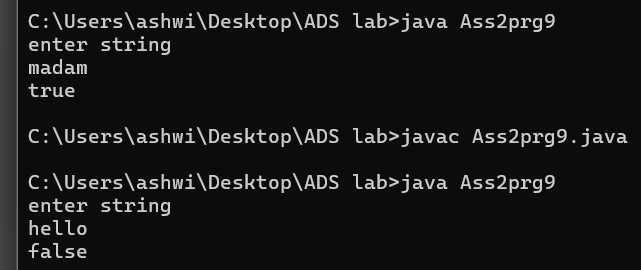
System.out.println("false");

}

}

}

**Output:**



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]

import java.util.Arrays;

public class Ass2prg10 {

public static void main(String[] args) {

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

int d1 = 2;

System.out.println("Input: arr = " + Arrays.toString(arr1) + ", d = " + d1);

rotateLeft(arr1, d1);

System.out.println("Output: " + Arrays.toString(arr1));

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

int d2 = 1;

System.out.println("\nInput: arr = " + Arrays.toString(arr2) + ", d = " + d2);

rotateLeft(arr2, d2);

System.out.println("Output: " + Arrays.toString(arr2));

}

public static void rotateLeft(int[] arr, int d) {

int n = arr.length;

int[] temp = new int[d];

// Store the first d elements in a temporary array

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

temp[i] = arr[i];

}

// Shift the remaining elements to the left

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

arr[i - d] = arr[i];

}

// Copy back the temporary array to the end

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

arr[n - d + i] = temp[i];

}

}

}

