**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 :

public class Pattern {

static void p(int n, int cr) {

if (cr > n) {

return ;

}

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

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

}

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

*p*(n, cr+1);

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

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

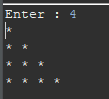
int n = sc.nextInt();

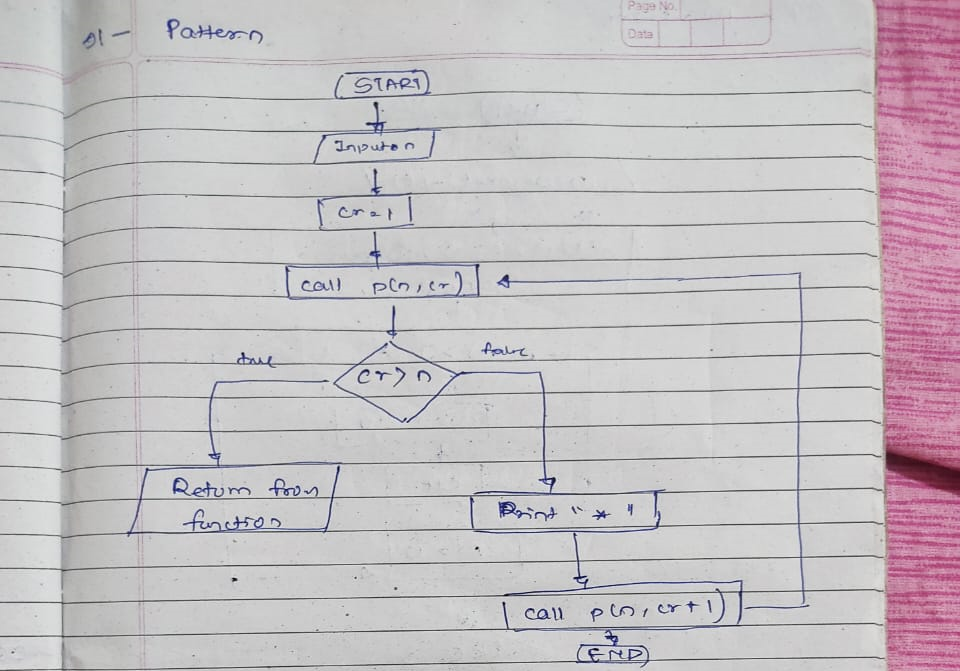
int cr = 1;

*p*(n, cr);

}

}





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

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

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

int arr1[] = new int[arr.length];

int j = 0, n = arr.length;

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

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

arr[j++] = arr[i];

}

}

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

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

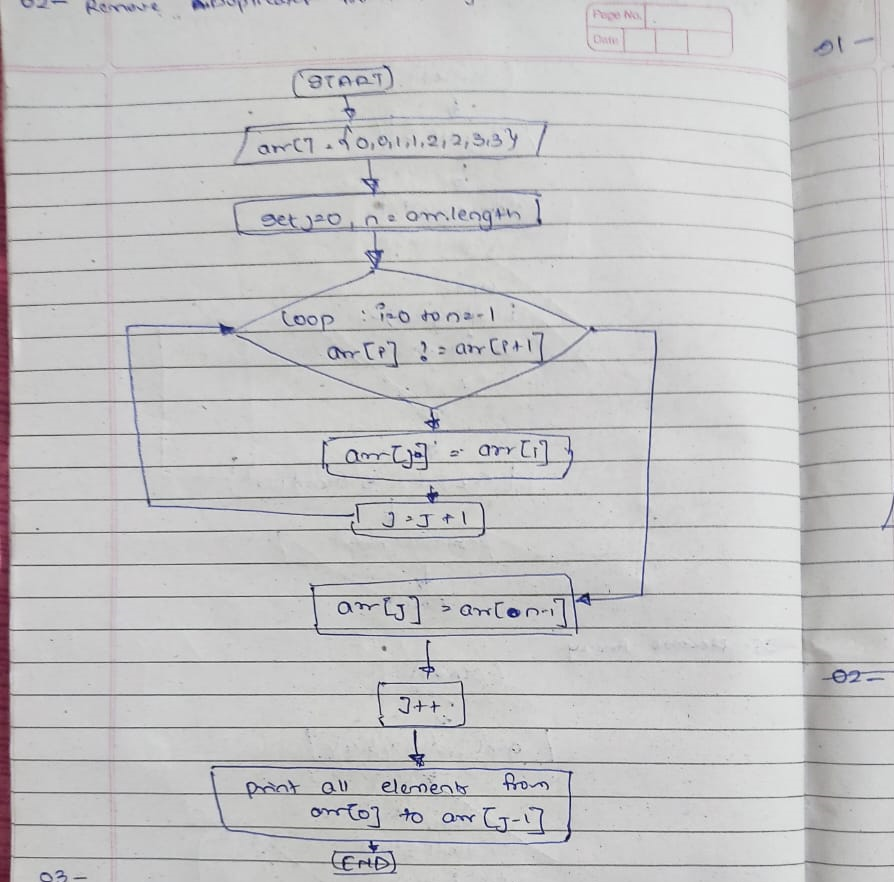
System.***out***.println(arr[i]);

}

}

}





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"

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("String : ");

String s = sc.nextLine();

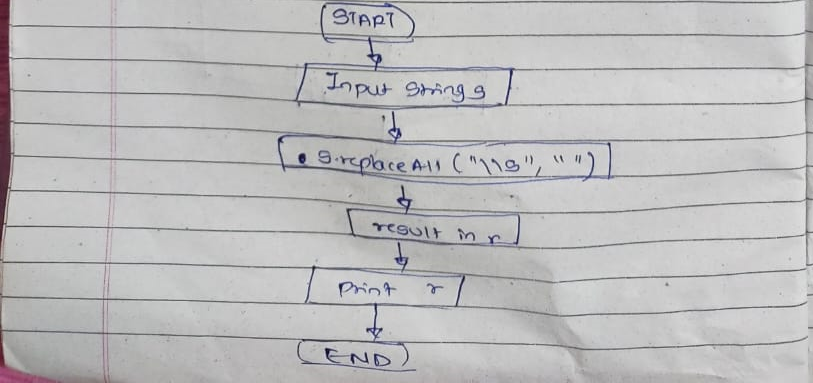
String r = s.replaceAll("\\s", "");

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

}

}





4. Reverse a String

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

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("String : ");

String s = sc.nextLine();

String rev = "" ;

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

rev += s.charAt(i);

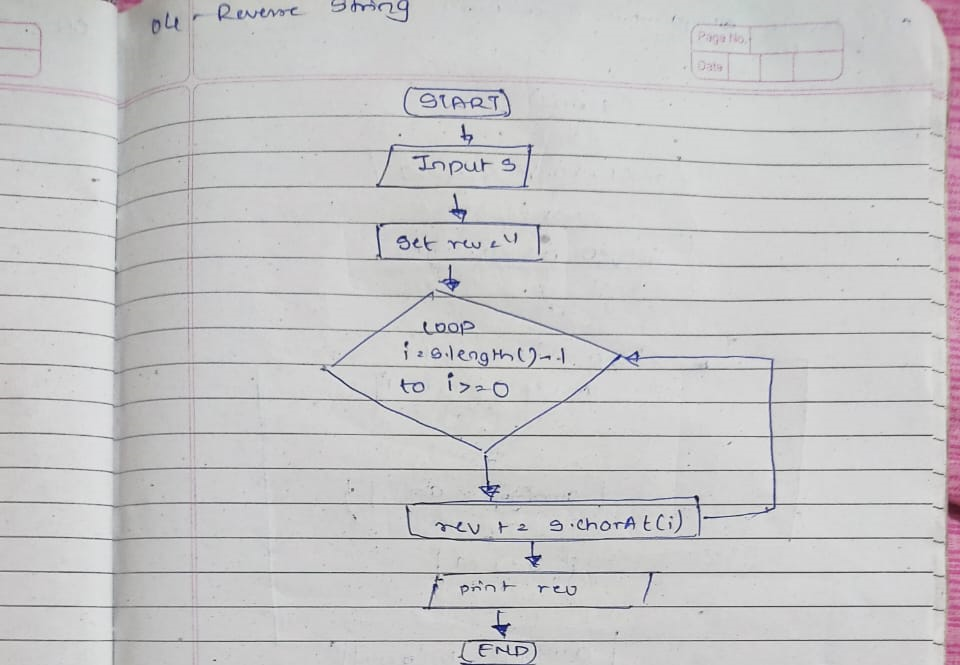
}

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

}

}





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]

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("String : ");

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

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

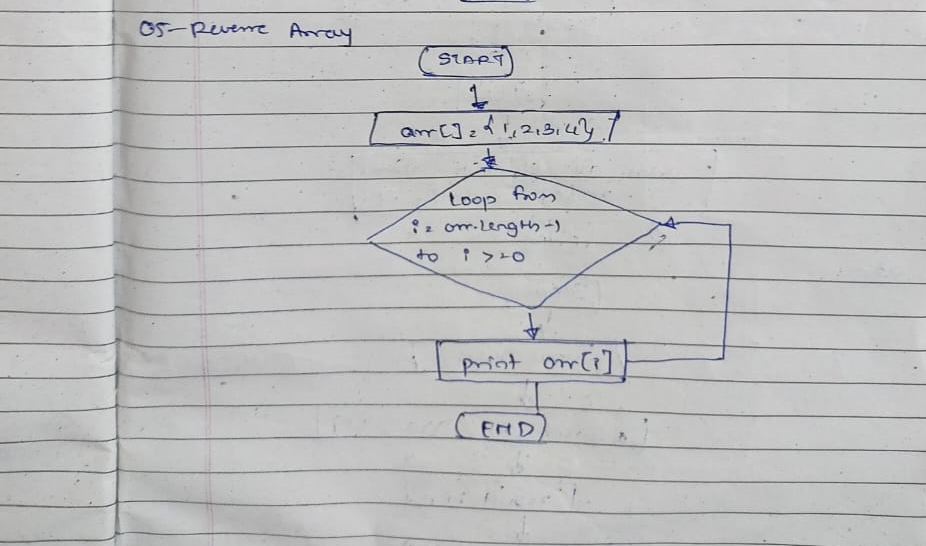
System.***out***.print(arr[i]);

}

}

}





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"

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("String : ");

String s = sc.nextLine();

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

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

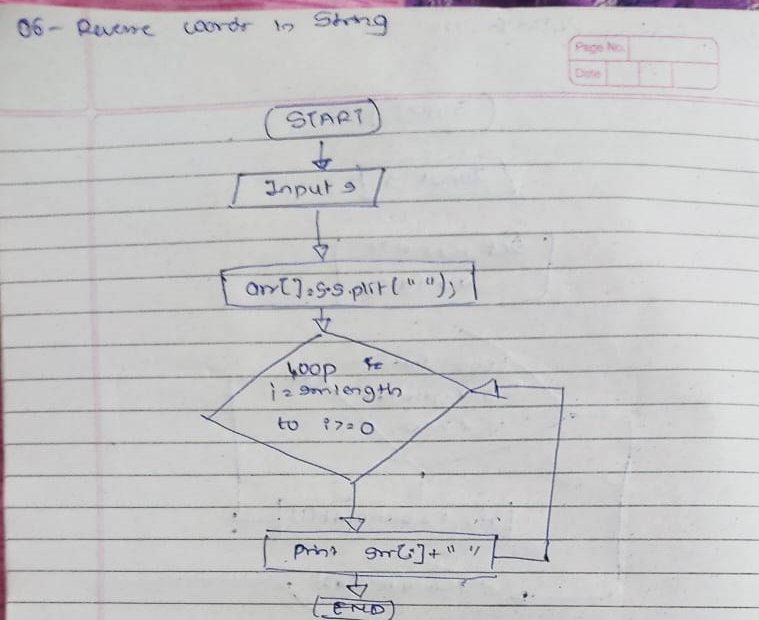
System.***out***.print(srr[i]+" ");

}

}

}





7. Reverse a Number

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

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

public class Remove {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("Int : ");

int n = sc.nextInt();

int count=0, rev=0;

int t= 0;

while ( n > 0 ) {

t= n%10;

rev = rev \* 10 + t;

n= n/ 10;

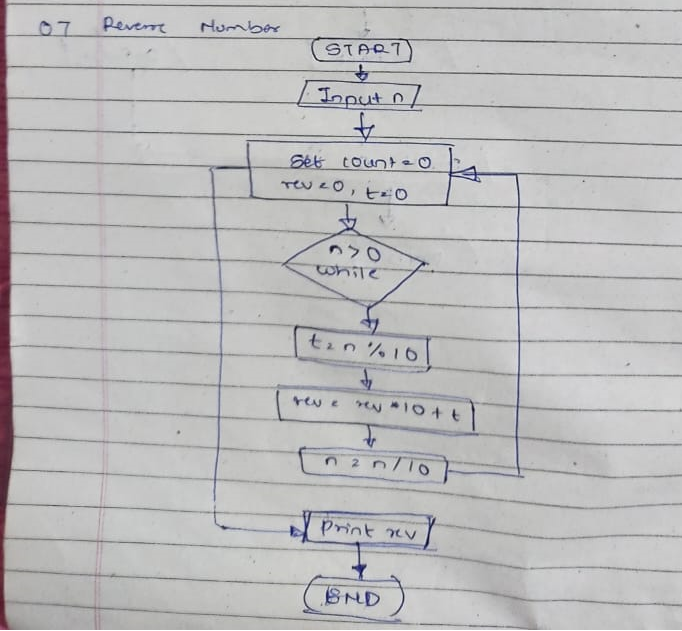
}

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

}

}





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:

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

System.***out***.print("String : ");

String s = sc.nextLine();

String rev = "";

boolean a = false;

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

rev += s.charAt(i);

}

if (rev.equals(s)) {

a = true;

}

else {

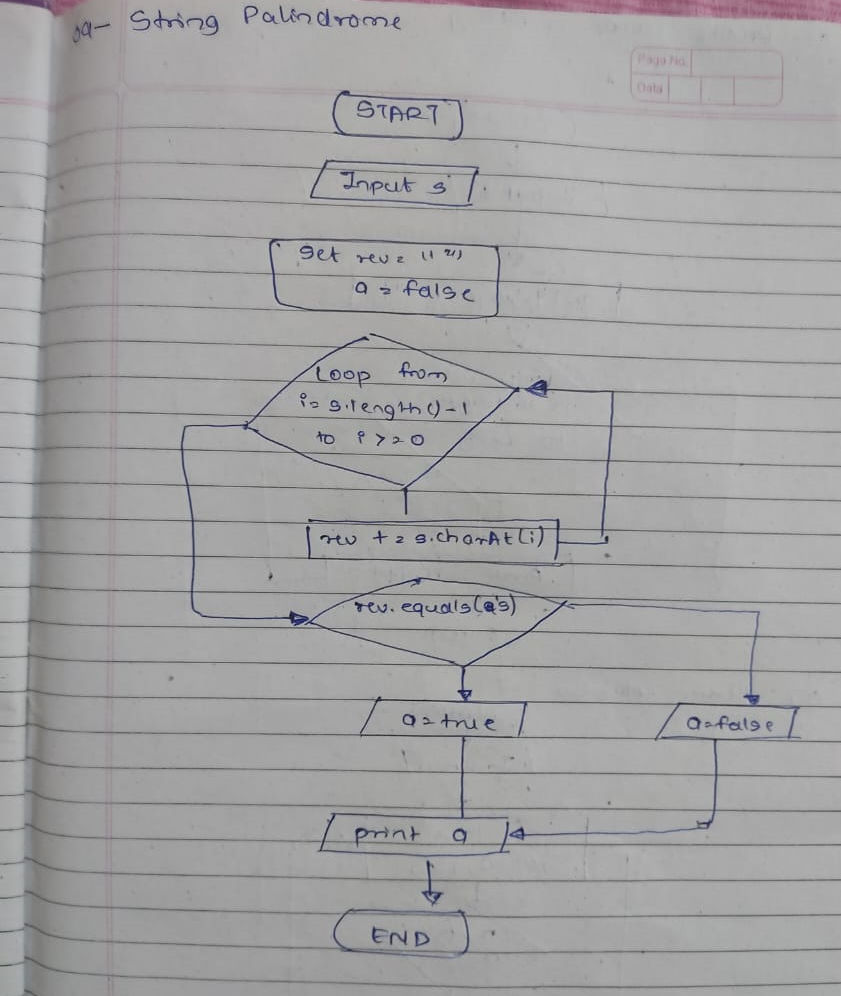
a = false;

}

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

}





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]

public static void main(String[] args) {

// **TODO** Auto-generated method stub

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

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

int n = sc.nextInt() ;

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

int first = arr[0], j;

for (j = 0; j < arr.length - 1; j++) {ec

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

}

arr[j] = first;

}

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

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

}

}



