**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**

**package** Assignment2;

**import** java.util.Scanner;

**public** **class** PrintPattern {

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

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

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

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

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

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

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

}

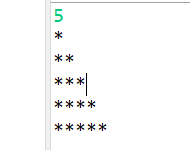
System.***out***.println();

}

}

}

**Output**

****

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**

**package** Assignment2;

**import** java.util.Scanner;

**public** **class** ArrayUniqueLength {

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

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

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

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

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

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

a[i]=sc.nextInt();

}

**int** newlen=1;

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

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

a[newlen]=a[i];

newlen++;

}

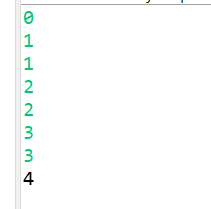
}

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

}

}

**Output**



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**

**package** Assignment2;

**import** java.util.Scanner;

**public** **class** RmvSpaceFromSring {

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

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

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

String s=sc.nextLine();

**char** a[]=s.toCharArray();

**for**(**char** c:a) {

**if**((c>='A' && c<='Z') ||(c>='a' && c<='z')) {

System.***out***.print(c);

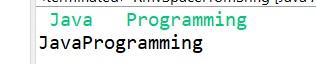
}

}

}

}

**Output**

****

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**

**package** Assignment2;

**import** java.util.Scanner;

**public** **class** StringReverse {

**static** String reverse(String s) {

**if**(s.length()==0) {

**return** s;

}

**return** (s.substring(1))+s.charAt(0);

}

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

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

String s=sc.nextLine();

String rev=*reverse*(s);

System.***out***.println(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]

**Program**

**package** Assignment2;

**public** **class** ReverseArray {

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

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

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

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

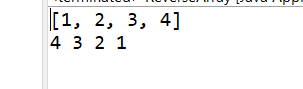
System.***out***.print(a[i]+" ");

}

}

}

**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**

**package** Assignment2;

**import** java.util.Arrays;

**import** java.util.Collection;

**import** java.util.Collections;

**import** java.util.List;

**public** **class** RevWordsString {

**public** **static** String reverseWordsInSentence(String sentence) {

// Split the sentence into words

String[] words = sentence.split(" ");

// Convert the array to a list for easy reversal

List<String> wordList = Arrays.*asList*(words);

// Reverse the order of words

Collections.*reverse*(wordList);

// Join the words back into a single string

**return** String.*join*(" ", wordList);

}

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

// Test case 1

String sentence1 = "Hello World";

String result1 = *reverseWordsInSentence*(sentence1);

System.***out***.println("Reversed: " + result1);

}

}

**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**

**package** Assignment2;

**import** java.util.Scanner;

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

**static** **int** revNum(**int** n,**int** temp) {

**if**(n==0)

**return** temp;

temp=temp\*10+n%10;

**return** *revNum*(n/10,temp);

}

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

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

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

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

**int** rev=*revNum*(n,0);

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

}

}

**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

**Program**

**Output**

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**

**package** Assignment2;

**import** java.util.Scanner;

**public** **class** StringPalindrome {

**static** String strPalindrome(String s) {

**if**(s.length()==0)

**return** s;

**return** *strPalindrome*(s.substring(1))+s.charAt(0);

}

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

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

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

String s=sc.nextLine();

String temp=s;

String pal=*strPalindrome*(s);

**boolean** rev=(temp.equals(pal))? **true**:**false**;

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

}

}

**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]

**Program**

**package** Assignment2;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** ArrayLeftRotation {

**static** **int**[] rotate(**int** a[],**int** d) {

**int** n=a.length;

d=d%n;

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

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

rotArray[i]=a[d+i];

}

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

rotArray[n-d+i]=a[i];

}

**return** rotArray;

}

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

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

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

**int** d=2;

**int** res[]=*rotate*(a,d);

System.***out***.println(Arrays.*toString*(res));

}

}

**Output**

