**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:

import java.util.\*;

public class Main

{

public static void main(String[] args) {

System.out.println("Enter the number of rows (n): ");

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();

}

}

}

**Explanation:**

1. The outer loop (for (int i = 1; i <= n; i++)) controls the number of rows.
2. The inner loop (for (int j = 1; j <= i; j++)) controls the number of stars printed on each row.
3. For each row, the number of stars printed is equal to the row number.

time complexity of O(n^2)

+-----------+

| Start |

+-----------+

|

+-----------+

| Input n |

+-----------+

|

+-------------------+

| i = 1 |

+-------------------+

|

+-----------------------+

| i <= n ? |

+-----------------------+

| | False

True +-----------+

| | End |

| +-----------+

+-----------------------+

| Initialize j = 1 |

+-----------------------+

|

+-----------------------+

| j <= i ? |

+-----------------------+

| | False

True

|

+-----------------------+

| Print "\*" |

+-----------------------+

|

+-----------------------+

| Increment j |

+-----------------------+

|

+-----------------------+

| Increment i |

+-----------------------+

|

+-----------------------+

| Print newline |

+-----------------------+

|

+-----------------------+

| Repeat outer loop |

+-----------------------+

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

Code:

public class Main {

public static int removeDuplicates(int[] arr)

{

if (arr.length == 0)

{

return 0 ;

}

int newLength = 1 ;

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

{

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

{

arr[newLength] = arr[i];

newLength++;

}

}

return newLength;

}

public static void main(String[] args)

{

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

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

System.out.println("New length of arr1 : " + removeDuplicates(arr1));

System.out.println("New length of arr2 : " + removeDuplicates(arr2));

}

}

**Explanation:**

1. The program starts by checking if the array is empty.
2. It initializes newLength to 1 since the first element is always included.
3. It iterates through the array, comparing each element with the previous one.
4. When a new element is found (not equal to the previous), it is added to the new position in the array.

**Time Complexity:** O(n)  
**Space Complexity:** O(1)

+-----------+

| Start |

+-----------+

|

+-----------+

| Input n |

+-----------+

|

+-------------------+

| Create array |

+-------------------+

|

+-------------------+

| newLength = 1 |

+-------------------+

|

+-----------------------+

| i = 1 |

+-----------------------+

|

+-----------------------+

| i < length ? |

+-----------------------+

| | False |

| True |

| | |

+-----------------------+

| Compare arr[i] |

| with arr[i - 1] |

+-----------------------+

| | No |

| | |

| | Yes |

| | |

+-----------------------+

| Update arr[newLength]|

| Increment newLength |

+-----------------------+

|

+-----------------------+

| Increment i |

+-----------------------+

|

+-----------------------+

| Repeat loop |

+-----------------------+

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"

Code:

package Monu;

import java.util.\*;

public class Whitespace {

    public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

System.out.println("Enter a String :");

String str = sc.nextLine();

String nowhite = str.replaceAll("\\s", "");

System.out.println("String Without Spcaes :"+nowhite);

}

}

A screenshot of a computer

Description automatically generated

4. Reverse a String

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

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

Code:

package Monu;

import java.util.\*;

public class Reverse2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a String :");

        String str = sc.nextLine();

char arr[] = new char[str.length()];

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

    arr[i] = str.charAt(i);

}

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

    System.out.println(arr[i]);

}

}

    }

A screenshot of a computer

Description automatically generated

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]

Code;

package Monu;

import java.lang.reflect.Array;

import java.util.\*;

public class Reversearr {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of elements you want in the array:");

        int n = sc.nextInt();

        sc.nextLine(); // Consume the newline character

        String arr[] = new String[n];  // Set the array size

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

            System.out.println("Enter a String at index " + i + ":");

            arr[i] = sc.nextLine();

        }

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

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

        }

    }

}

A screenshot of a computer

Description automatically generated

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"

Code;

package Monu;

import java.util.\*;

public class Reverse2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a String :");

        String str = sc.nextLine();

char arr[] = new char[str.length()];

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

    arr[i] = str.charAt(i);

}

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

    System.out.println(arr[i]);

}

}

    }

A screenshot of a computer

Description automatically generated

7. Reverse a Number

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

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

Code:

package Monu;

import java .util.\*;

public class Reverse3 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter an Integer : ");

        int n = sc.nextInt();

     int reversed = 0;

     while(n!=0){

       reversed = reversed \* 10 + n% 10;

       n= n/10;

     }

     System.out.println("Reversed Number: " + reversed);

    }

}

A screenshot of a computer

Description automatically generated

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:

Code:

package Monu;

import java.util.Scanner;

public class Palindrome {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter a string:");

        String str = sc.nextLine();

        String reversed = new StringBuilder(str).reverse().toString();

        if (str.equals(reversed)) {

            System.out.println("The given string is a palindrome.");

        } else {

            System.out.println("The given string is not a palindrome.");

        }

    }

}

A screenshot of a computer program

Description automatically generated

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]

Code:

package Monu;

import java.util.\*;

public class Rotation {

    public static void main(String[] args) {

        Integer[] arr = {1, 2, 3, 4, 5, 6, 7};

        int n = 2; // Rotate by 2 positions

       List<Integer> list = Arrays.asList(arr);

       Collections.rotate(list, -n);

       arr = list.toArray(new Integer[0]);

       System.out.println(Arrays.toString(arr));

   }

}

A screenshot of a computer

Description automatically generated