

# ADS Ass 2 Solution

Monday, September 30, 2024 9:02 AM

## 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:

```
*  
**  
***  
****  
*****
```

Main.java

Share

Run

```
1- import java.util.*;  
2  
3  
4- class StarPattern {  
5-     public static void main(String[] args) {  
6         int n;  
7         Scanner sc = new Scanner (System.in);  
8         System.out.println("Enter n : ");  
9         n = sc.nextInt ();  
10  
11-         for (int i = 1; i<=n; i++ ) {  
12-             for (int j = 1; j<=i; j++) {  
13                 System.out.print ("*");  
14             }  
15             System.out.println ();  
16         }  
17  
18     }  
19 }
```

Output

```
java -cp /tmp/yaamHItxsM/StarPattern  
Enter n :  
3  
*  
**  
***  
  
=== Code Execution Successful ===
```

```
import java.util.*;  
  
class StarPattern {  
    public static void main(String[] args) {  
        int n;  
        Scanner sc = new Scanner (System.in);  
        System.out.println("Enter n : ");  
        n = sc.nextInt ();  
  
        for (int i = 1; i<=n; i++ ) {  
            for (int j = 1; j<=i; j++) {  
                System.out.print ("*");  
            }  
            System.out.println (" ");  
        }  
    }  
}
```

Main.java	Output
<pre> 1- import java.util.*; 2 3 4- class StarPattern { 5-     public static void main(String[] args) { 6         int n; 7         Scanner sc = new Scanner (System.in); 8         System.out.println("Enter n : "); 9         n = sc.nextInt (); 10 11-         for (int i = 1; i&lt;=n; i++ ) { 12-             for (int j = 1; j&lt;=i; j++) { 13                 System.out.print ("*"); 14             } 15             System.out.println (); 16         } 17 18     } 19 } </pre>	<pre> java -cp /tmp/96geqyFISN/StarPattern Enter n : 5 * ** *** **** *****  === Code Execution Successful === </pre>

## 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

Main.java	Output
<pre> 1- class DuplicateArrayRemoval { 2 3-     public int removeDuplicates(int arr[]) { 4         if (arr.length == 0) 5             return 0; 6 7         int j = 0; 8         for (int i = 1; i &lt; arr.length; i++) { // Start from i = 1,            since arr[0] is unique by default 9-             if (arr[j] != arr[i]) { 10                 arr[++j] = arr[i]; 11             } 12         } 13 14         return j + 1; // Length of unique elements 15     } 16 17-     public static void main(String[] args) { 18         DuplicateArrayRemoval arrm = new DuplicateArrayRemoval(); 19         int arr[] = {1, 1, 2}; 20         int uniqueCount = arrm.removeDuplicates(arr); 21 22         // Print the number of unique elements 23         System.out.println("Number of unique elements: " +            uniqueCount); 24     } </pre>	<pre> java -cp /tmp/qycQqbWDup/DuplicateArrayRemoval Number of unique elements: 2  === Code Execution Successful === </pre>

```

class DuplicateArrayRemoval {

    public int removeDuplicates(int arr[]) {
        if (arr.length == 0)
            return 0;

        int j = 0;
        for (int i = 1; i < arr.length; i++) { // Start from i = 1, since arr[0] is unique by default
            if (arr[j] != arr[i]) {
                arr[++j] = arr[i];
            }
        }

        return j + 1; // Length of unique elements
    }

    public static void main(String[] args) {
        DuplicateArrayRemoval arrm = new DuplicateArrayRemoval();
        int arr[] = {1, 1, 2};
        int uniqueCount = arrm.removeDuplicates(arr);
    }
}

```

```

// Print the number of unique elements
System.out.println("Number of unique elements: " + uniqueCount);
}
}

```

### 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"

Main.java	Output
<pre> 1 class StringSpaceRemoval { 2 3 public static void main(String[] args) { 4     String str1 = "Hello World"; 5     String str2 = "Java Programming"; 6 7     String new1 = str1.replace(" ", ""); 8     System.out.println(new1); 9 10    String new2 = str2.replace(" ", ""); 11    System.out.println(new2); 12 } 13 } 14 </pre>	<pre> java -cp /tmp/Ri69VgqqC1/StringSpaceRemoval HelloWorld JavaProgramming  === Code Execution Successful === </pre>

```

class StringSpaceRemoval {

    public static void main(String[] args) {
        String str1 = "Hello World";
        String str2 = "Java Programming";

        String new1 = str1.replace(" ", "");
        System.out.println(new1);

        String new2 = str2.replace(" ", "");
        System.out.println(new2);
    }
}

```

### 4. Reverse a String

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

Test Cases:

Input: "hello"

Output: "olleh"

Input: "Java"

Output: "avaJ"

Main.java	Output
<pre> 1 // Java Program to Reverse a Word 2 3 import java.io.*; 4 import java.util.Scanner; 5 6 class GFG { 7     public static void main (String[] args) { 8 9         String str= "hello", nstr="";    //nstr is empty string to            hold reverse string 10        char ch; 11 12        System.out.print("Original word: "); 13        System.out.println("hello"); //Example word 14 15 16 17        for (int i=0; i&lt;str.length(); i++) 18        { 19            ch= str.charAt(i); //extracts each character 20            nstr= ch+nstr; //adds each character in front of the                 existing string 21        } 22        System.out.println("Reversed word: "+ nstr); 23 24    } </pre>	<pre> java -cp /tmp/1H5onxoctv/GFG Original word: hello Reversed word: olleh  === Code Execution Successful === </pre>

  

```

// Java Program to Reverse a Word

import java.io.*;
import java.util.Scanner;

class GFG {
    public static void main (String[] args) {

        String str= "hello", nstr="";    //nstr is empty string to hold reverse string
        char ch;

        System.out.print("Original word: ");
        System.out.println("hello"); //Example word

        for (int i=0; i<str.length(); i++)
        {
            ch= str.charAt(i); //extracts each character
            nstr= ch+nstr; //adds each character in front of the existing string
        }
        System.out.println("Reversed word: "+ nstr);

    }
}

```

### 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]

```

import java.util.Arrays;

class ReverseArrayInPlace {

    // Method to reverse the array in place
    public static void reverseArray(int[] arr) {
        int start = 0;
        int end = arr.length - 1;

        while (start < end) {
            // Swap the elements at start and end
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;

            // Move towards the middle
            start++;
            end--;
        }
    }

    public static void main(String[] args) {
        // Test case 1
        int[] arr1 = {1, 2, 3, 4};
        System.out.println("Original array: " + Arrays.toString(arr1));
        reverseArray(arr1);
    }
}

```

```

java -cp /tmp/wfLsLYoF04/ReverseArrayInPlace
Original array: [1, 2, 3, 4]
Reversed array: [4, 3, 2, 1]
Original array: [7, 8, 9]
Reversed array: [9, 8, 7]

```

```

=== Code Execution Successful ===

```

```

public static void main(String[] args) {
    // Test case 1
    int[] arr1 = {1, 2, 3, 4};
    System.out.println("Original array: " + Arrays.toString(arr1));
    reverseArray(arr1);
    System.out.println("Reversed array: " + Arrays.toString(arr1));

    // Test case 2
    int[] arr2 = {7, 8, 9};
    System.out.println("Original array: " + Arrays.toString(arr2));
    reverseArray(arr2);
    System.out.println("Reversed array: " + Arrays.toString(arr2));
}
}

```

```

import java.util.Arrays;

class ReverseArrayInPlace {

    // Method to reverse the array in place
    public static void reverseArray(int[] arr) {
        int start = 0;
        int end = arr.length - 1;

        while (start < end) {
            // Swap the elements at start and end
            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;

            // Move towards the middle
            start++;
            end--;
        }
    }

    public static void main(String[] args) {
        // Test case 1
    }
}

```

```

int[] arr1 = {1, 2, 3, 4};
System.out.println("Original array: " + Arrays.toString(arr1));
reverseArray(arr1);
System.out.println("Reversed array: " + Arrays.toString(arr1));

// Test case 2
int[] arr2 = {7, 8, 9};
System.out.println("Original array: " + Arrays.toString(arr2));
reverseArray(arr2);
System.out.println("Reversed array: " + Arrays.toString(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"

```

import java.util.Scanner;
class ReverseWordsInString {
    // Method to reverse the words in a string
    public static String reverseWords(String sentence) {
        // Split the sentence into words
        String[] words = sentence.split(" ");
        StringBuilder reversedSentence = new StringBuilder();

        // Iterate through the array of words in reverse order
        for (int i = words.length - 1; i >= 0; i--) {
            reversedSentence.append(words[i]);
            // Add space after each word (except for the last one)
            if (i != 0) {
                reversedSentence.append(" ");
            }
        }
        return reversedSentence.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Test case 1
        System.out.println("Enter a sentence:");
        String sentence1 = sc.nextLine();
        String result1 = reverseWords(sentence1);
        System.out.println("Reversed words: " + result1);
    }
}

```

```

java -cp /tmp/nv4dRsmHNQ/ReverseWordsInString
Enter a sentence:
Hello World
Reversed words: World Hello

=== Code Execution Successful ===

```

```

import java.util.Scanner;
class ReverseWordsInString {
    // Method to reverse the words in a string
    public static String reverseWords(String sentence) {
        // Split the sentence into words
        String[] words = sentence.split(" ");
        StringBuilder reversedSentence = new StringBuilder();

        // Iterate through the array of words in reverse order
        for (int i = words.length - 1; i >= 0; i--) {
            reversedSentence.append(words[i]);
            // Add space after each word (except for the last one)
            if (i != 0) {
                reversedSentence.append(" ");
            }
        }
        return reversedSentence.toString();
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Test case 1
        System.out.println("Enter a sentence:");
        String sentence1 = sc.nextLine();
        String result1 = reverseWords(sentence1);
        System.out.println("Reversed words: " + result1);
    }
}

```

```
}  
}
```

#### 7. Reverse a Number

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

Test Cases:

Input: 12345

Output: 54321

Input: -9876

Output: -6789

```
class ReverseNumberUsingStringBuilder {  
  
    public static int reverseNumber(int num) {  
        // Convert the number to a string and handle negative  
        // numbers  
        String str = Integer.toString(Math.abs(num));  
  
        // Reverse the string using StringBuilder  
        String reversedStr = new StringBuilder(str).reverse  
            ().toString();  
  
        // Convert the reversed string back to an integer and  
        // restore the sign  
        int reversed = Integer.parseInt(reversedStr);  
        return num < 0 ? -reversed : reversed; // Apply the  
        // original sign  
    }  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter a number to reverse:");  
        int number = sc.nextInt();  
        int reversedNumber = reverseNumber(number);  
  
        System.out.println("Reversed number: " + reversedNumber);  
    }  
}
```

```
java -cp /tmp/rqQigTC5hb/ReverseNumberUsingStringBuilder
```

```
Enter a number to reverse:
```

```
12345
```

```
Reversed number: 54321
```

```
=== Code Execution Successful ===
```

```
import java.util.Scanner;
```

```
class ReverseNumberUsingStringBuilder {  
  
    public static int reverseNumber(int num) {  
        // Convert the number to a string and handle negative numbers  
        String str = Integer.toString(Math.abs(num));  
  
        // Reverse the string using StringBuilder  
        String reversedStr = new StringBuilder(str).reverse().toString();  
  
        // Convert the reversed string back to an integer and restore the sign  
        int reversed = Integer.parseInt(reversedStr);  
        return num < 0 ? -reversed : reversed; // Apply the original sign  
    }  
  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        System.out.println("Enter a number to reverse:");  
        int number = sc.nextInt();  
        int reversedNumber = reverseNumber(number);  
  
        System.out.println("Reversed number: " + reversedNumber);  
    }  
}
```

#### 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:

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]