Subject: Algorithm and Data Structure
Assignment 2

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 –
n=3

```
public class StarPattern {
    public static void main(String[] args) {
        int n = 3;
        printPattern(n);
    }

    public static void printPattern(int n) {
        if (n == 0) {
            return;
        }
        printPattern(n - 1);
        printStars(n);
        System.out.println();
    }

    public static void printStars(int n) {
        if (n == 0) {
            return;
        }
        System.out.print("*");
        printStars(n - 1);
    }
}
```

n=5

```
public class StarPattern {
    public static void main(String[] args) {
        int n = 5;
        printPattern(n);
    }

    public static void printPattern(int n) {
        if (n == 0) {
            return;
        }
        printPattern(n - 1);
        printStars(n);
        System.out.println();
    }

    public static void printStars(int n) {
        if (n == 0) {
            return;
        }
        System.out.print("*");
        printStars(n - 1);
    }
}
```

Output –

```
C:\Users\CSH\Desktop\ADS\Assignment 2>javac StarPattern.java

C:\Users\CSH\Desktop\ADS\Assignment 2>java StarPattern
*
**
***

C:\Users\CSH\Desktop\ADS\Assignment 2>javac StarPattern.java

C:\Users\CSH\Desktop\ADS\Assignment 2>java StarPattern
*
**
***
****
*****
```

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:

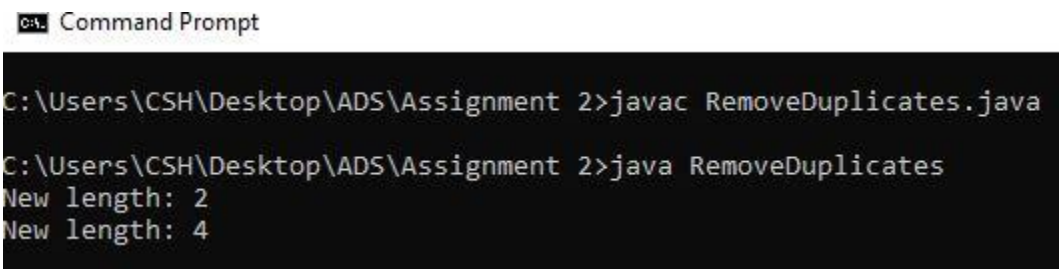
Input: arr = [1, 1, 2]
Output: 2
Input: arr = [0, 0, 1, 1, 2, 2, 3, 3]
Output: 4

Code –

```
public class RemoveDuplicates {

   public static void main(String[] args) {


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

      int newLength1 = removeDuplicates(arr1);

      System.out.println("New length: " + newLength1);


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

      int newLength2 = removeDuplicates(arr2);
```

```
        System.out.println("New length: " + newLength2);
    }

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

        int uniqueIndex = 0;

        for (int i = 1; i < arr.length; i++) {
            if (arr[i] != arr[uniqueIndex]) {
                uniqueIndex++;
                arr[uniqueIndex] = arr[i];
            }
        }

        return uniqueIndex + 1;
    }
}
```

Output –



3. Remove White Spaces from String
Problem: Write a Java program to remove all white spaces from a given string.

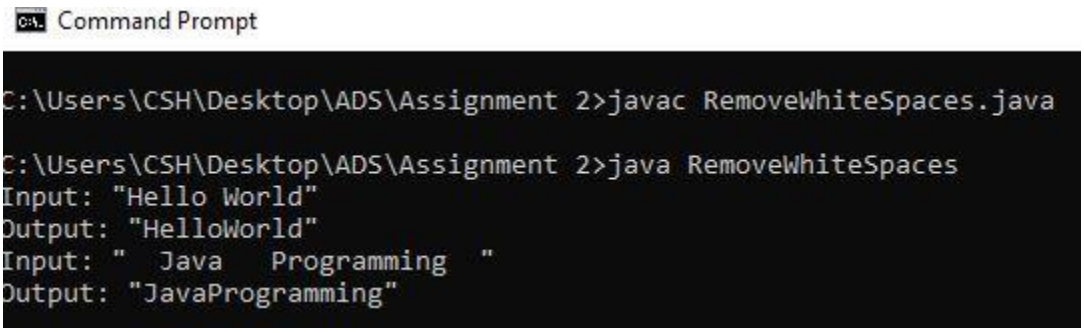
Test Cases:

Input: "Hello World"
Output: "HelloWorld"
Input: " Java   Programming "
Output: "JavaProgramming"

Code –

```
public class RemoveWhiteSpaces {
   public static String removeSpaces(String input) {

      return input.replaceAll("\\s", "");
   }

   public static void main(String[] args) {

      String input1 = "Hello World";
      System.out.println("Input: \"" + input1 + "\"");
      System.out.println("Output: \"" + removeSpaces(input1) + "\"");

      String input2 = "  Java   Programming  ";
      System.out.println("Input: \"" + input2 + "\"");
      System.out.println("Output: \"" + removeSpaces(input2) + "\"");
   }
}
```

Output –



4. Reverse a String
Problem: Write a Java program to reverse a given string.

Test Cases:

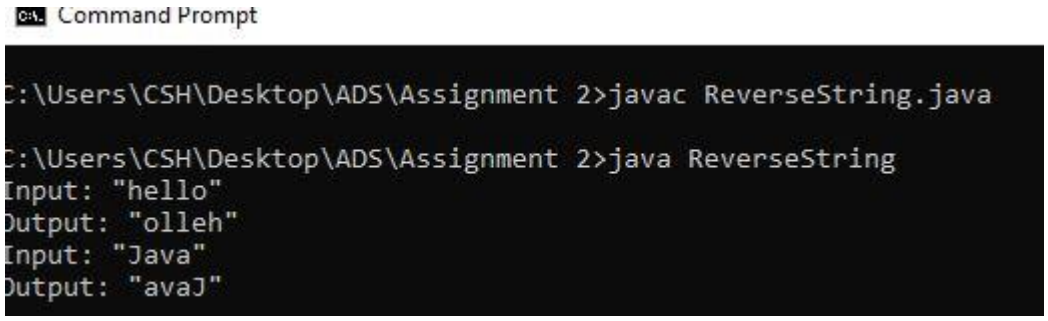
Input: "hello"
Output: "olleh"
Input: "Java"
Output: "avaJ"

Code –

```
public class ReverseString {

   public static String reverse(String input) {

      String reversed = "";


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

         reversed += input.charAt(i);

      }

      return reversed;

   }


   public static void main(String[] args) {


      String input1 = "hello";

      System.out.println("Input: \"" + input1 + "\"");
```

```
        System.out.println("Output: \"" + reverse(input1) + "\"");


        String input2 = "Java";
        System.out.println("Input: \"" + input2 + "\"");
        System.out.println("Output: \"" + reverse(input2) + "\"");
    }
}
```

Output –



5. Reverse Array in Place
Problem: Write a Java program to reverse an array in place.

Test Cases:

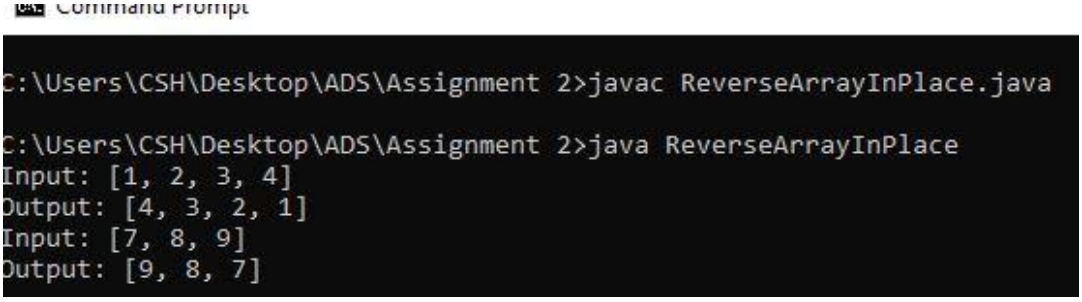
Input: arr = [1, 2, 3, 4]
Output: [4, 3, 2, 1]
Input: arr = [7, 8, 9]
Output: [9, 8, 7]

Code –

```
import java.util.Arrays;


public class ReverseArrayInPlace {
    public static void reverseArray(int[] arr) {
        int n = arr.length;
```

```
        for (int i = 0; i < n / 2; i++) {

            int temp = arr[i];
            arr[i] = arr[n - i - 1];
            arr[n - i - 1] = temp;
        }
    }

    public static void main(String[] args) {

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

        int[] arr2 = {7, 8, 9};
        System.out.println("Input: " + Arrays.toString(arr2));
        reverseArray(arr2);
        System.out.println("Output: " + Arrays.toString(arr2));
    }
}
```

Output –



6. Reverse Words in a String
Problem: Write a Java program to reverse the words in a given sentence.

Test Cases:

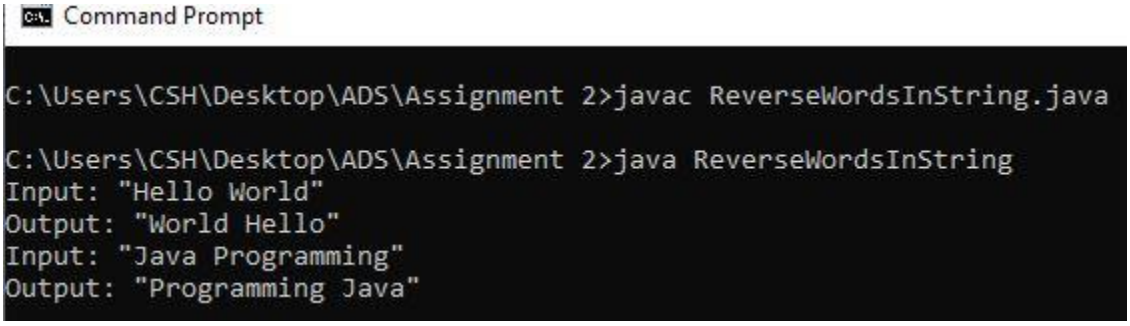
Input: "Hello World"
Output: "World Hello"
Input: "Java Programming"
Output: "Programming Java"

Code –

```
public class ReverseWordsInString {
    public static String reverseWords(String input) {

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

        String reversedSentence = "";

        for (int i = words.length - 1; i >= 0; i--) {
            reversedSentence += words[i];
            if (i != 0) {
                reversedSentence += " ";
            }
        }

        return reversedSentence;
    }

    public static void main(String[] args) {

        String input1 = "Hello World";
        System.out.println("Input: \"" + input1 + "\"");
        System.out.println("Output: \"" + reverseWords(input1) + "\"");
```

```
        String input2 = "Java Programming";
        System.out.println("Input: \"" + input2 + "\"");
        System.out.println("Output: \"" + reverseWords(input2) + "\"");
    }
}
```

Output –



7. Reverse a Number
Problem: Write a Java program to reverse a given number.

Test Cases:

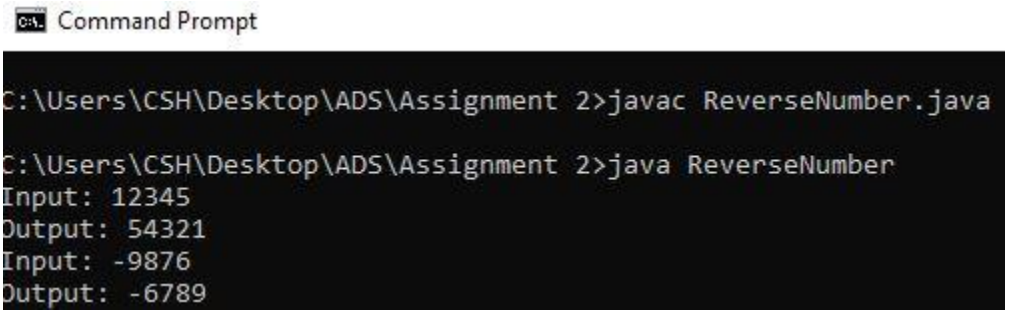
Input: 12345
Output: 54321
Input: -9876
Output: -6789

Code –

```
public class ReverseNumber {
    public static int reverseNumber(int num) {
        int reversed = 0;
        int a = num < 0 ? -1 : 1;

        num = Math.abs(num);
```

```
        while (num > 0) {
            int lastDigit = num % 10;
            reversed = reversed * 10 + lastDigit;
            num /= 10;
        }

        return reversed * a;
    }

    public static void main(String[] args) {

        int input1 = 12345;
        System.out.println("Input: " + input1);
        System.out.println("Output: " + reverseNumber(input1));

        int input2 = -9876;
        System.out.println("Input: " + input2);
        System.out.println("Output: " + reverseNumber(input2));
    }
}
```

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:

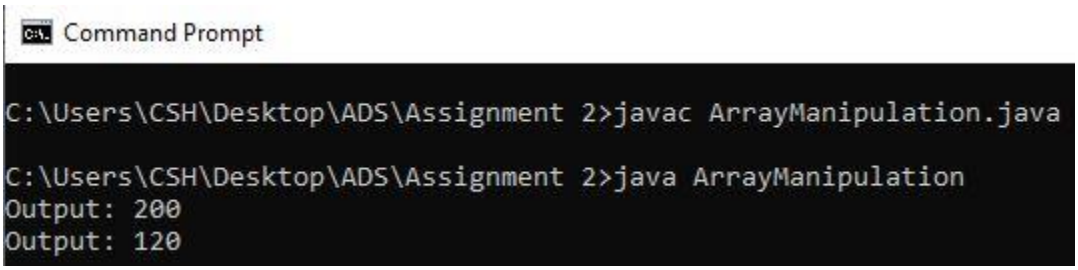
Input: n = 5, queries = [[1, 2, 100], [2, 5, 100], [3, 4, 100]]
Output: 200
Input: n = 4, queries = [[1, 3, 50], [2, 4, 70]]
Output: 120

Code –

```
public class ArrayManipulation {

   public static long arrayManipulation(int n, int[][] queries) {


      long[] arr = new long[n + 1];


      for (int[] query : queries) {

         int a = query[0];

         int b = query[1];

         int c = query[2];


         arr[a - 1] += c;

         if (b < n) {

            arr[b] -= c;

         }

      }


      long max = Long.MIN_VALUE;

      long currentSum = 0;


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

         currentSum += arr[i];
```

```
            if (currentSum > max) {
                max = currentSum;
            }
        }

        return max;
    }

    public static void main(String[] args) {

        int[][] queries1 = {{1, 2, 100}, {2, 5, 100}, {3, 4, 100}};
        int n1 = 5;
        System.out.println("Output: " + arrayManipulation(n1, queries1));

        int[][] queries2 = {{1, 3, 50}, {2, 4, 70}};
        int n2 = 4;
        System.out.println("Output: " + arrayManipulation(n2, queries2));
    }
}
```

Output –



9. String Palindrome
Problem: Write a Java program to check if a given string is a palindrome.

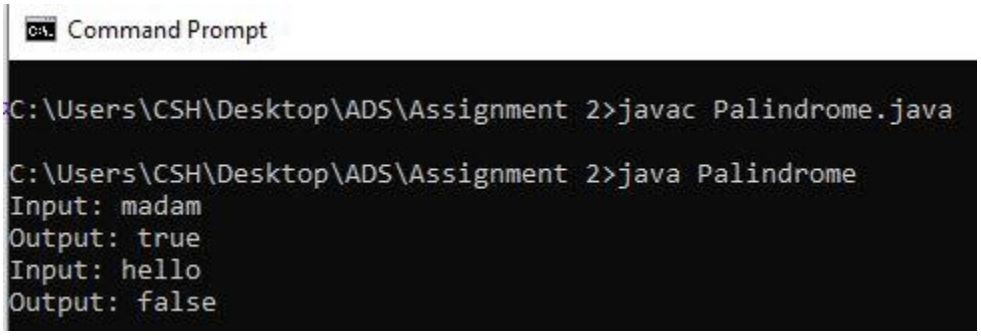
Test Cases:

Input: "madam"
Output: true
Input: "hello"
Output: false

Code –

```
public class Palindrome {
   public static boolean isPalindrome(String str) {

      StringBuilder reversedStr = new StringBuilder(str).reverse();

      return str.equals(reversedStr.toString());
   }

   public static void main(String[] args) {

      String input1 = "madam";
      System.out.println("Input: " + input1);
      System.out.println("Output: " + isPalindrome(input1));

     String input2 = "hello";
      System.out.println("Input: " + input2);
      System.out.println("Output: " + isPalindrome(input2));
   }
}
```

Output –



10. Array Left Rotation
Problem: Write a Java program to rotate an array to the left by d positions.

Test Cases:

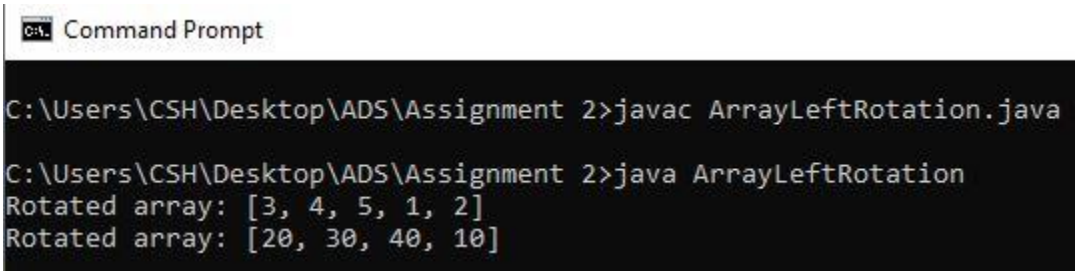
Input: arr = [1, 2, 3, 4, 5], d = 2
Output: [3, 4, 5, 1, 2]
Input: arr = [10, 20, 30, 40], d = 1
Output: [20, 30, 40, 10]

Code –

```
import java.util.Arrays;


public class ArrayLeftRotation {


   public static void leftRotate(int[] arr, int d) {
      int n = arr.length;


      d = d % n;


      reverse(arr, 0, d - 1);


      reverse(arr, d, n - 1);


      reverse(arr, 0, n - 1);
   }
```

```
    private static void reverse(int[] arr, int start, int end) {
        while (start < end) {

            int temp = arr[start];
            arr[start] = arr[end];
            arr[end] = temp;
            start++;
            end--;
        }
    }

    public static void main(String[] args) {

        int[] arr1 = {1, 2, 3, 4, 5};
        int d1 = 2;
        leftRotate(arr1, d1);
        System.out.println("Input: " [1, 2, 3, 4, 5], d = 2);
        System.out.println("Output: " + Arrays.toString(arr1));

        int[] arr2 = {10, 20, 30, 40};
        int d2 = 1;
        leftRotate(arr2, d2);
        System.out.println("Input: " [10, 20, 30, 40], d = 1);
                    System.out.println("Output: " + Arrays.toString(arr2))
    }
}
```

Output –


Command Prompt
C:\Users\CSH\Desktop\ADS\Assignment 2>javac ArrayLeftRotation.java
C:\Users\CSH\Desktop\ADS\Assignment 2>java ArrayLeftRotation
Rotated array: [3, 4, 5, 1, 2]
Rotated array: [20, 30, 40, 10]