Day-5

Quiz-2

1. Given an array . Create two arrays one for Odd Elements and other for Even Elements.
Input: [10,3,5,12,17,22]
Output:
[10,12,22]
[3,5,7]

CODE:

```
import java.util.ArrayList;
import java.util.Scanner;
public class SeparateOddEvenArrays {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of elements in the array: ");
      int n = scanner.nextInt();

      int[] inputArray = new int[n];
      System.out.println("Enter the elements of the array:");

      for (int i = 0; i < n; i++) {
         inputArray[i] = scanner.nextInt();
      }
      ArrayList<Integer> evenArray = new ArrayList<>();
      ArrayList<Integer> oddArray = new ArrayList<>();

      for (int num : inputArray) {
         if (num % 2 == 0) {
            evenArray.add(num);
         } else {
            oddArray.add(num);
         }
      }
      int[] evenElements = evenArray.stream().mapToInt(Integer::intValue).toArray();
      int[] oddElements = oddArray.stream().mapToInt(Integer::intValue).toArray();
      System.out.print("Even Elements: [");
      for (int i = 0; i < evenElements.length; i++) {
         System.out.print(evenElements[i]);
         if (i < evenElements.length - 1) {
            System.out.print(",");
         }
      }
      System.out.println("]");
      System.out.print("Odd Elements: [");
      for (int i = 0; i < oddElements.length; i++) {
         System.out.print(oddElements[i]);
         if (i < oddElements.length - 1) {
```

```
            System.out.print(",");
        }
    }
    System.out.println("]");
    scanner.close();
  }
}
```

OUTPUT:

```
Enter the number of elements in the array: 6
Enter the elements of the array:10
3
5
12
17
22
Even Elements: [10,12,22]
Odd Elements: [3,5,17]
```

CODE:

```
import java.util.Scanner;
public class StringCompression {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a string: ");
      String input = scanner.nextLine();
      String compressedString = compressString(input);
      System.out.println("Original String: " + input);
      System.out.println("Compressed String: " + compressedString);
      scanner.close();
   }
   private static String compressString(String input) {
      StringBuilder compressed = new StringBuilder();
      int count = 1;
     for (int i = 0; i < input.length(); i++) {
         if (i < input.length() - 1 && input.charAt(i) == input.charAt(i + 1)) {
            count++;
         } else {
            compressed.append(input.charAt(i));
            if (count > 1) {
               compressed.append(count);
            }
            count = 1;
         }
      }
```

```
  return compressed.toString();
    }
}
```

OUTPUT:

```
Enter a string: AAABBC
Original String: AAABBC
Compressed String: A3B2C
```

```
Enter a string: AAABBCCCDE
Original String: AAABBCCCDE
Compressed String: A3B2C3DE
```

3.
Input : zohocorporationteam
Output :

```
z o h o c o r
          p
        o
     r
  a
 t
i o n t e a m
```

CODE:

```
import java.util.Scanner;
public class ZigZagPattern {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        System.out.print("Enter the number of rows: ");
        int numRows = scanner.nextInt();
        printZigZagPattern(input, numRows);
        scanner.close();
    }
  private static void printZigZagPattern(String input, int numRows) {
       if (numRows <= 0) {
            System.out.println("Number of rows should be greater than 0.");
            return;
        }
        if (numRows == 1 || numRows >= input.length()) {
            System.out.println(input);
            return;
        }

        StringBuilder[] zigzag = new StringBuilder[numRows];
```

```
        for (int i = 0; i < numRows; i++) {
            zigzag[i] = new StringBuilder();
        }

        int row = 0;
        boolean goingDown = false;

        for (char ch : input.toCharArray()) {
            zigzag[row].append(ch);
            if (row == 0 || row == numRows - 1) {
                goingDown = !goingDown;
            }
            row += goingDown ? 1 : -1;
        }
        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j < zigzag[i].length(); j++) {
                char currentChar = zigzag[i].charAt(j);
                System.out.print((currentChar == '\u0000') ? " " : currentChar + " ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT: