1.Concatenate the string 1

```
Import java.util.Scanner;
Public class Solution {
   Public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String s1 = sc.next();
      String s2 = sc.next();
      String result = s1 + s2;
      System.out.println(result);
      Sc.close();
   }
}
```

2.Count the vowels 7

```
Import java.util.Scanner;
Public class VowelCounter {
   Public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String input = sc.nextLine();
      Sc.close();
      Int count = 0;
      For (int I = 0; I < input.length(); i++) {
```

```
        Char ch = Character.toLowerCase(input.charAt(i));
        If (ch == ‘a’ || ch == ‘e’ || ch == ‘I’ || ch == ‘o’ || ch == ‘u’) {
          Count++;
        }
      }
      System.out.println(count);
   }
}
```

3.Count each character in the string 1

```
Import java.util.*;
Public class Solution {
   Public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String input = sc.nextLine();
      Sc.close();
      LinkedHashMap<Character, Integer> charCount = new
LinkedHashMap<>();
      For (int I = 0; I < input.length(); i++) {
        Char ch = input.charAt(i);
        charCount.put(ch, charCount.getOrDefault(ch, 0) + 1);
      }
```

```
      For (Map.Entry<Character, Integer> entry : charCount.entrySet())
{
         System.out.println(entry.getKey() + “   “ + entry.getValue());
      }
   }
}
```

4.Count vowels, consonants, digits, special characters

```
Import java.util.Scanner;
Public class Main {
   Public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String str = sc.nextLine();
      Int vowels = 0;
      Int consonants = 0;
      Int digits = 0;
      Int specialChars = 0;
      Str = str.toLowerCase();
      For (char ch : str.toCharArray()) {
         If (ch >= ‘a’ && ch <= ‘z’) {
            If (ch == ‘a’ || ch == ‘e’ || ch == ‘I’ || ch == ‘o’ || ch == ‘u’) {
               Vowels++;
```

```
            } else {
                Consonants++;
            }
        } else if (ch >= '0' && ch <= '9') {
            Digits++;
        } else {
            specialChars++;
        }
    }
    System.out.println("vowels:" + vowels);
    System.out.println("consonants:" + consonants);
    System.out.println("digits:" + digits);
    System.out.println("special characters:" + specialChars);
  }
}
```

5.Check if string contains only digits 2

```
Import java.io.*;
Import java.util.*;
Public class Solution {
  Public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
```

```
        String s = scanner.nextLine();
        Scanner.close();
        Boolean allDigits = true;
        For (char c : s.toCharArray()) {
            If (!Character.isDigit©) {
                allDigits = false;
                break;
            }
        }
        If (allDigits) {
            System.out.println(“only digits”);
        } else {
            System.out.println(“no”);
        }
    }
}
```

6.String anagram 6

```
Import java.io.*;
Import java.util.*;
Public class Solution {
    Public static void main(String[] args) {
```

```
        Scanner scan = new Scanner(System.in);
        String a = scan.next();
        String b = scan.next();
        Scan.close();
        A = a.toLowerCase();
        B = b.toLowerCase();
        If (a.length() != b.length()) {
           System.out.println(“The given strings are not an anagram”);
           Return;
        }
        Char[] charArrayA = a.toCharArray();
        Char[] charArrayB = b.toCharArray();
        Arrays.sort(charArrayA);
        Arrays.sort(charArrayB);
        If (Arrays.equals(charArrayA, charArrayB)) {
           System.out.println(“The given strings are an anagram”);
        } else {
           System.out.println(“The given strings are not an anagram”);
        }
    }
}
```

7.Alternating Code 3

```
Import java.io.*;
Import java.util.*;
Public class Solution {
   Public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String s = sc.next();
      Sc.close();
      Boolean isAlternating = true;

      If (s.length() > 1 && s.charAt(0) == s.charAt(1)) {
         isAlternating = false;
      } else {
         For (int I = 0; I < s.length() – 2; i++) {
            If (s.charAt(i) != s.charAt(I + 2)) {
               isAlternating = false;
               break;
            }
         }
      }
```

```
        If (isAlternating) {
            System.out.println(“Yes”);
        } else {
            System.out.println(“No”);
        }
    }
}
```