Problem Statement 1: “Given a string, check if the string is palindrome or not.” A string is said to be palindrome if the reverse of the string is the same as the string

```
import java.util.Scanner;
public class palindrome
{
   public static void main(String[] args)
          {
      int count = 0;
      Scanner sc = new Scanner(System.in);
      System.out.print("enter the string: ");
      String p = sc.nextLine();
      char c[] = p.toCharArray();
      for (int i=0,j=c.length-1; i<(c.length)/2; i++,j--)
          {
        if(c[i]!=c[j])
                    {
          count++;
          break;
          }
      }
      if(count>0)
          {
        System.out.println("not a palindrome");
      }
      else
          {
        System.out.println("Palindrome");
      }
   }
}
```

Problem Statement 2:Given a string, write a program to count the number of vowels, consonants, and spaces in that string

```
import java.util.*;
public class countvcs{
   public static void main(String[] args){
      int vowel=0, consonant=0, whitespace =0;
      System.out.print("enter string: ");
      Scanner sc = new Scanner(System.in);
      String s = sc.nextLine();
      for (int i = 0; i < s.length(); i++){
         char c = s.charAt(i);
    if (c == 'a'||c== 'e' || c== 'i' || c== 'o' || c== 'u' || c== 'A' || c== 'E' || c== 'I' || c== 'O' ||c == 'U'){
            vowel++;
         }
         else if(c == ' '){
            whitespace++;
         }
         else{
            consonant++;
         }
      }
      System.out.print("vowel="+vowel+"\n consonant:"+consonant+"\n whitespace:"+whitespace);
   }
}
```

Problem Statement 3: Given a String, write a program to remove vowels from a given String.

```
import java.util.*;
public class removewhitespace{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String s = sc.nextLine();
      String result = "";
      for(int i=0;i<s.length();i++){
         char c = s.charAt(i);
         if(c != ' '){
            result += c;
      }
   }
   System.out.print(result);
}
}
```

Problem Statement 4: Given a string, write a program to remove all the whitespaces from the string.

```
import java.util.*;
public class removevowel{
    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter string: ");
        String s = sc.nextLine();
        String result = "";
         for(int i=0;i<s.length();i++){
           char c = s.charAt(i);
   if(!(c == 'a'||c== 'e' || c== 'i' || c== 'o' || c== 'u' || c== 'A' || c== 'E' || c== 'I' || c== 'O' ||c == 'U'))
{
              result += c;
        }
    }
    System.out.print(result);
}
}
```

Problem Statement 5 : Write a program to remove all characters from a string except alphabets in a given string

```
import java.util.*;
public class removeeverythingexceptalpha
{
   public static void main(String[] args)
{
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String s = sc.nextLine();
      String result ="";
      for(int i=0;i<s.length();i++)
          {
         char c = s.charAt(i);
         if(c>=65 && c<=90 || c>=97 && c<=122)
          {
            result += c;
         }
      }
   System.out.print(result);
}
}
```

Problem Statement 6: Reverse a String. Write a program that reverses a given string. Problem: Given a string, calculate the sum of numbers in a string (multiple consecutive digits are considered one number)

```
import java.util.*;


public class sum {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String s = sc.nextLine();


      String num = "";
      int sum1 = 0;


      for (int i = 0; i < s.length(); i++) {
         char c = s.charAt(i);


         if (Character.isDigit(c)) {
            num += c;
         } else if (!num.isEmpty()) {
            int num1 = Integer.parseInt(num);
            sum1 += num1;
            num = "";
         }
      }
      if (!num.isEmpty()) {
         int num1 = Integer.parseInt(num);
         sum1 += num1;
      }
      System.out.println("Sum of numbers in the string: " + sum1);
   }
}
```

Problem Statement 7: Given a string, write a program to Capitalize the first and last character of each word of that string.

```
import java.util.*;
public class capital{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String s = sc.nextLine();
      char c[] = s.toCharArray();
      for(int i=0;i<s.length()-1;i++){
         char c1 = c[i];
         if(i==0 || c[i+1] == ' '){
            c[i]=Character.toUpperCase(c1);
         }
         else if(c[i-1]== ' '){
            c[i]=Character.toUpperCase(c1);
         }
      }
      c[s.length()-1]=Character.toUpperCase(c[s.length()-1]);
      s = new String(c);
   System.out.print(s);
}
}
```

Problem Statement 8: Given two strings, check if two strings are anagrams of each other or not.

```
import java.util.*;
public class anagram{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string 1: ");
      String s1 = sc.nextLine();
      System.out.print("Enter string 2: ");
      String s2 = sc.nextLine();
      if(s1.length() == s2.length()) {
         char c1 []= s1.toCharArray();
         char c2 []= s2.toCharArray();
         Arrays.sort(c1);
         Arrays.sort(c2);
         s1 = new String(c1);
         s2 = new String(c2);
         if(s1.equals(s2)){
            System.out.print("Anagram!");
         }
         else{
            System.out.println("Not Anagram!");
         }
      }
      else{
         System.out.print("Not Anagram!");
      }
}
}
```

Problem Statement 9: Given a String, find the largest word in the string.

```
import java.util.*;
public class largestword{
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter string: ");
      String s = sc.nextLine();
      String words []= s.split(" ");
      int max =words[0].length();
      String largestword = words[0];
      for(int i=1;i<words.length;i++){
         if(max<words[i].length()){
            largestword = words[i];
            max = words[i].length();
         }

      }
      System.out.println(largestword+" is the largest word in the string");
}
}
```