Assignment 4

1. Palindrome Check

Problem: Check if a string is a palindrome.

Input Format: A single string (no spaces)

Output Format: "Palindrome" or "Not Palindrome"

Constraints: 1 <= length <= 1000

Sample Input:

madam

Sample Output:

Palindrome

Program:

```
import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String str = sc.nextLine();
      String reversed = "";
      for (int i = str.length() - 1; i >= 0; i--) {
         reversed += str.charAt(i);
      }
      if (str.equals(reversed)) {
         System.out.println("Palindrome");
      } else {
         System.out.println("Not Palindrome");
     }}}
```

2. Count Vowels and Consonants

Problem: Count the number of vowels and consonants in the input string.

Input Format: A single line containing a string (may have spaces)

Output Format: Vowels: X, Consonants: Y

Constraints: Input will only contain alphabetic characters and spaces

Sample Input:

Java Programming

Sample Output:

Vowels: 5, Consonants: 10

Program:

```
import java.util.Scanner;

public class Main {

  public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);

    String str = sc.nextLine().toLowerCase();

    int v=0,c=0;

    for(int i=0i<str.length();i++){

      char ch=str.charAt(i);

      if(ch.isLetter(ch)){

        if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){

          v++;

        }

        c++;

      }

    }}}
```

3. Remove Duplicates

Problem: Remove all duplicate characters from a string while preserving order.

Input Format: A single string

Output Format: A string with duplicates removed

Constraints: Lowercase letters only

Sample Input:

programming

Sample Output:

Progamin

```
import java.util.Scanner;
public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String s1 = sc.nextLine();
      String result = "";
      for (int i = 0; i < s1.length(); i++) {
         char ch = s1.charAt(i);
         if (result.indexOf(ch) == -1) {
            result += ch;
         }
      }
       System.out.println(result);}}
```

4. Reverse a String Using StringBuilder

Problem: Reverse the input string using StringBuilder.

Input Format: A single string

Output Format: Reversed string

Constraints: 1 <= length <= 1000

Sample Input:

Hello

Sample Output:

olleH

Program

```
import java.util.Scanner;

public class Main {

   public static void main(String[] args) {

      Scanner sc = new Scanner(System.in);

      String s1 = sc.nextLine();

      StringBuilder sb=new StringBuilder(s1);

      sb.reverse();

      System.out.println(sb.toString());

   }

}
```

5. Anagram Check

Problem: Check if two strings are anagrams of each other.

Input Format: Two strings on separate lines

Output Format: "Anagrams" or "Not Anagrams"

Constraints: Only lowercase letters, no spaces

Sample Input:

listen\nsilent

Sample Output:

Anagrams

Program

```
import java.util.Arrays;
import java.util.Scanner;
public class Main {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    String str1 = sc.nextLine();
    String str2 = sc.nextLine();
    if (areAnagrams(str1, str2)) {
      System.out.println("Anagrams");
    } else {
      System.out.println("Not Anagrams");
    }
  }
  public static boolean areAnagrams(String s1, String s2) {
    if (s1.length() != s2.length()) {
      return false;
    }
    char[] arr1 = s1.toCharArray();
```

```
        char[] arr2 = s2.toCharArray();
        Arrays.sort(arr1);
        Arrays.sort(arr2);
        return Arrays.equals(arr1, arr2);
    }
}
```

6. Capitalize First Letter of Each Word

Problem: Capitalize the first letter of every word in a sentence.

Input Format: A sentence (multiple words)

Output Format: Sentence with each word capitalized

Constraints: Input contains only lowercase letters and spaces

Sample Input:

java is fun

Sample Output:

Java Is Fun

Program:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String line = sc.nextLine();
        String result = "";
        String[] words = line.split(" ");
```

```
        for (String word : words) {
            result += word.substring(0, 1).toUpperCase() + word.substring(1) + " ";
        }
        System.out.println(result.trim());
    }
}
```

7. Count Word Occurrences

Problem: Count how many times a given word appears in the sentence.

Input Format: First line: Sentence
Second line: Word to search

Output Format: An integer (count)

Constraints: Case-sensitive match

Sample Input:

Java is simple. Java is powerful.\nJava

Sample Output:

2

Program

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String sentence = sc.nextLine();
        String word = sc.nextLine();
```

```
        String[] words = sentence.split(" ");
        int count = 0;
        for (String w : words) {
            if (w.equals(word)) {
                count++;
            }
        }
        System.out.println(count);
    }
}
```

8. Toggle Case

Problem: Convert lowercase letters to uppercase and vice versa.

Input Format: A single string

Output Format: The toggled string

Constraints: Input contains only alphabetic characters

Sample Input:

HeLLo

Sample Output:

hEllO

Program:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
```

```
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        String res = "";
        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (c >= 'A' && c <= 'Z') {
                res += (char)(c + 32);
            } else {
                res += (char)(c - 32);
            }
        }
        System.out.println(res);
    }
}
```