1. Write a C++ program that accepts both a string and a single character from the user. The program should determine how many times the character is contained in the string. (Hint: Search the string by using the find(str, int) function. This function should be used in a loop that starts the index value at 0 and then changes the index value to 1 past the index of where the char was last found.)

2. Write a C++ program that counts the number of words in a string. A word is encountered whenever a transition from a blank space to a nonblank character is encountered. The string contains only words separated by blank spaces.(Hint: use gets() to get a string with spaces.)

3. Enter a string and print whether the string is a palindrome. A palindrome is a string that reads the same backwards and forwards.

**Input format.**

Input as a string (no whitespace characters in the string, string length less than 100).

**Output format.**

If the string is a palindrome, print yes; Otherwise, output no.

**Input1:**

abcdedcba

**Output1:**

yes

**Input2:**

agg

**Output2:**

no

4. Count the number of occurrences of each letter of the input string, case – insensitive.

**Input format.**

The input is a string containing only 26 English letters, and the length of the string is not more than 100.

**Output format.**

Output the number of occurrences of each letter.

**Input:**

Supercaliocious

**Output:**

the number of a : 1

the number of c : 2

the number of e : 1

the number of i : 2

the number of l : 1

the number of o : 2

the number of p : 1

the number of r : 1

the number of s : 2

the number of u : 2

5. Count the number of occurrences of a substring in a string.

**Input format.**

The length of the input string should not exceed 100, and the substring should not exceed the original string.

**Output format.**

The number of the substring.

**input**.

I have a pen, I have an apple, Apple,pen.

apple

**output.**

The number of substrings is 1