

## **LAB :- 6 (Strings)**

### **1.Vowels vs Consonants**

Write a program to input T strings (S) from user and print count of vowels and consonants in it.

**Input:**

2

List

Apple

**Output:**

1 3

2 3

### **2. Length of String - II**

You have a string (A). You have to print length of input string.

**Input:**

Python

**Output:**

6

### **3.Is is Palindrome?**

Write a program to input T strings (S) from user and print 1 if it is palindrome otherwise print 0. NOTE:A string is palindrome if it reads the same from backward as from forward.

**Input:**

3

abcba

axax

abba

**Output:**

1

0

1

#### **4.Trim From Ends**

You are given a character string A. You to trim(remove) both leading and trailing asterisk characters('\*') in the string and return the resultant string.

**Input:**

```
A = "***h*e*l*l*o**"
```

**Output:**

```
h*e*l*l*o
```

#### **5.Reverse string**

Write a program to reverse the words present in a string. Everything else should be preserved. Check example input/output. Note: There are no punctuation and special characters in the string.

The string will only contain alphanumeric characters and spaces.

**Input:**

Everyone loves data science

**Output:**

enoyrevE sevol atad ecneics

#### **6.Reverse the word**

You are given lowercase string (A) and you have to print after reversing that.

**Input:**

String

**Output:**

gnirtS

#### **7.First Occurrence**

You are given a character string A, having length N and an integer ASCII code B.

You have to tell the leftmost occurrence of the character having ASCII code equal to B, in A or report that it does not exist.

**Input:**

A = "aabbcc"

B = 98

**Output:**

2

### **8.First Occurrence Of Word**

You are given two character strings A and B.

You have to find the first occurrence of string B in string A, as a substring, and return the starting position of first occurrence.

A substring is a contiguous sequence of characters within a string. For e.g "at" is a substring in "catalogue".

**Input:**

A = "aabababaa"

B = "ba"

**Output:**

2

### **9. tolower()**

You are given a function to\_lower() which takes a character array A as an argument.

Convert each character of A into lowercase characters if it exists. If the lowercase of a character does not exist, it remains unmodified.

The uppercase letters from A to Z are converted to lowercase letters from a to z respectively.

Print the lowercase version of the given character array.

**Input:**

A = ["S" , "u", "y", "A", "s" , "H"]

**Output:**

["s" , "u", "y", "a", "s" , "h"]

### **10.toupper()**

You are given a function to\_upper() consisting of a character array A.

Convert each character of A into Uppercase character if it exists. If the Uppercase of a character does not exist, it remains unmodified.

The lowercase letters from a to z is converted to uppercase letters from A to Z respectively.

Print the uppercase version of the given character array.

#### **Input:**

A = ["s" , "U", "y", "A", "s" , "H"]

#### **Output:**

["S" , "U", "Y", "A", "S" , "H"]

### **11.Isalnum()**

You are given a function isalpha() consisting of a character array A.

Print 1 if all the characters of a character array are alphanumeric (a-z, A-Z, and 0-9) else, print 0.

A = ["P" , "y", "t", "h", "O" , "n", "2" , "4"]

#### **Output:**

1

A = ["P" , "y", "t", "h", "O" , "n", "2" , "4"]

### **12.Isalpha()**

You are given a function isalpha() consisting of a character array A.

Print 1 if all the characters of the character array are alphabetical (a-z and A-Z), else print 0.