

Assignment No.12

#Python Program to Replace all Occurrences of 'a' with \$ in a String

```
def replace_a_with_dollar(input_string):
```

```
    result = ""
```

```
    for char in input_string:
```

```
        if char == 'a':
```

```
            result += '$'
```

```
        else:
```

```
            result += char
```

```
    return result
```

```
text = "banana and apple"
```

```
modified_text = replace_a_with_dollar(text)
```

```
print("Original String:", text)
```

```
print("Modified String:", modified_text)
```

#Python Program to Remove the nth Index Character from a Non-Empty String

```
def remove_nth_char(input_string, n):
```

```
    result = ""
```

```
    index = 0
```

```
    for char in input_string:
```

```
        if index != n:
```

```
            result += char
```

```
        index += 1
```

```
    return result
```

```
text = "computer"
```

```
n = 3
```

```
if n >= 0 and n < len(text):
```

```
    new_text = remove_nth_char(text, n)
```

```
    print("Original String:", text)
```

```
    print("Modified String:", new_text)
else:
    print("Invalid index")
```

#Python Program to Detect if Two Strings are Anagrams

```
def checkanagram(str1, str2):
    if(len(str1) == len(str2)):
        data = {}
        for chr1, chr2 in zip(str1, str2):
            if chr1 not in data.keys():
                data[chr1] = 1
            else:
                data[chr1] = data[chr1] + 1
            if chr2 not in data.keys():
                data[chr2] = -1
            else:
                data[chr2] = data[chr2] - 1
        for val in data.values():
            if(val != 0):
                return f"'{str1}' and '{str2}' are not anagram string"
            else:
                return f"'{str1}' and '{str2}' are anagram string"
    else:
        return f"'{str1}' and '{str2}' are not anagram string"

str1 = "listen"
str2 = "silent"

print(checkanagram(str1, str2))
```

#Python Program to Form a New String where the First Character and the Last Character have been Exchanged

```
def string_length(s):  
    count = 0  
    for _ in s:  
        count += 1  
    return count  
  
def swap_first_last(s):  
    length = string_length(s)  
    if length <= 1:  
        return s  
    result = ""  
    first_char = "  
    last_char = "  
    index = 0  
    for char in s:  
        if index == 0:  
            first_char = char  
        elif index == length - 1:  
            last_char = char  
        index += 1  
    index = 0  
    for char in s:  
        if index == 0:  
            result += last_char  
        elif index == length - 1:  
            result += first_char  
        else:  
            result += char  
        index += 1
```

```
    return result

input_str = "python"

output_str = swap_first_last(input_str)

print("Original String:", input_str)

print("Modified String:", output_str)
```

#Python Program to Count the Number of Vowels in a String

```
def is_vowel(ch):

    return ch == 'a' or ch == 'e' or ch == 'i' or ch == 'o' or ch == 'u' or \

        ch == 'A' or ch == 'E' or ch == 'I' or ch == 'O' or ch == 'U'

def count_vowels(s):

    count = 0

    for char in s:

        if is_vowel(char):

            count += 1

    return count

input_str = "Hello World"

vowel_count = count_vowels(input_str)

print("Input String:", input_str)

print("Number of Vowels:", vowel_count)
```

#Python Program to Take in a String and Replace Every Blank Space with Hyphen

```
def replace_space_with_hyphen(s):

    result = ""

    for char in s:

        if char == ' ':

            result += '-'

        else:

            result += char

    return result
```

```
input_str = "this is a test string"
output_str = replace_space_with_hyphen(input_str)
print("Original String:", input_str)
print("Modified String:", output_str)
```

#Python Program to Calculate the Length of a String Without Using a Library Function

```
def string_length(s):
    count = 0
    for _ in s:
        count += 1
    return count

input_str = "Hello, World!"
length = string_length(input_str)
print("Input String:", input_str)
print("Length of String:", length)
```

#Python Program to Remove the Characters of Odd Index Values in a String

```
def remove_odd_index_chars(s):
    result = ""
    index = 0
    for char in s:
        if index % 2 == 0:
            result += char
        index += 1
    return result

input_str = "abcdefg"
output_str = remove_odd_index_chars(input_str)
print("Original String:", input_str)
print("Modified String:", output_str)
```

#Python Program to Calculate the Number of Words and the Number of Characters Present in a String Function to calculate number of characters

```
def count_characters(s):  
    count = 0  
    for _ in s:  
        count += 1  
    return count  
  
def count_words(s):  
    in_word = False  
    word_count = 0  
    for char in s:  
        if char != ' ' and not in_word:  
            word_count += 1  
            in_word = True  
        elif char == ' ':  
            in_word = False  
    return word_count  
  
input_str = "This is a simple Python program"  
char_count = count_characters(input_str)  
word_count = count_words(input_str)  
print("Input String:", input_str)  
print("Number of Characters:", char_count)  
print("Number of Words:", word_count)
```

#Python Program to Take in Two Strings and Display the Larger String without Using Built-in Functions

```
def string_length(s):  
    count = 0  
    for _ in s:  
        count += 1
```

```

    return count
def larger_string(str1, str2):
    len1 = string_length(str1)
    len2 = string_length(str2)
    if len1 > len2:
        return str1
    elif len2 > len1:
        return str2
    else:
        return "Both strings are of equal length."
string1 = "hello"
string2 = "worldwide"
result = larger_string(string1, string2)
print("Larger String:", result)

```

#Python Program to replace every blank space with hyphen in a string.

```

def replace_space_with_hyphen(s):
    result = ""
    for char in s:
        if char == ' ':
            result += '-'
        else:
            result += char
    return result
input_str = "Python is fun and powerful"
output_str = replace_space_with_hyphen(input_str)
print("Original String:", input_str)
print("Modified String:", output_str)

```

#Python Program to count number of lowercase characters in a string

```

def count_lowercase(s):
    count = 0
    for ch in s:
        if 'a' <= ch <= 'z':
            count += 1
    return count

input_str = "Hello World! python 123"
print("Input String:", input_str)
print("Number of Lowercase Characters:", count_lowercase(input_str))

```

#Python Program to count number of digits and letters in a string

```

def count_letters_digits(s):
    letters = 0
    digits = 0
    for ch in s:
        if ('a' <= ch <= 'z') or ('A' <= ch <= 'Z'):
            letters += 1
        elif '0' <= ch <= '9':
            digits += 1
    return letters, digits

text = "Hello123World456"
letters, digits = count_letters_digits(text)
print("Letters:", letters)
print("Digits:", digits)

```

#Python Program to count the occurrences of each word in a string

```

def count_words(s):
    words = s.split()
    counts = {}
    for word in words:

```



```
    if word in counts:
        counts[word] += 1
    else:
        counts[word] = 1
for word in counts:
    print(word, ":", counts[word])
text = "this is a test this is only a test"
print("Input String:", text)
print("Word Occurrences:")
count_words(text)
```

#Python Program to find larger string without using built-in functions.

```
def get_length(s):
    count = 0
    for _ in s:
        count += 1
    return count

def find_larger_string(s1, s2):
    len1 = get_length(s1)
    len2 = get_length(s2)
    if len1 > len2:
        return s1
    elif len2 > len1:
        return s2
    else:
        return "Both strings are of equal length."

string1 = "Python"
string2 = "Programming"
print("String 1:", string1)
```

```
print("String 2:", string2)
```

```
print("Larger String:", find_larger_string(string1, string2))
```