Experiment No 01: Introduction to NLP String Processing

Rebecca Dias
Roll no: 18 BE CMPN A2
Pid: 182027

Aim: To perform the various processing techniques on string in Python

String Entries

4 cells hidden

Usage of Built-in Functions in String

44 cells hidden

Programming Exercises

1. Write a Python program to calculate the length of a string.

```
def calc_len(string):
  return len(string)


print(calc_len(string))
```

```
19
```

2. Write a Python program to count the number of characters in a string.

```
def count_char(string):
  return len(string.replace(" ",""))

print(count_char(string))
```

```
16
```

3. Write a Python program to get a string made of the first 2 and the last 2 chars from a given a string. If the string length is less than 2, return instead of the empty string.

```
def join_f_l_2(string):
  if len(string) < 2:
    return
  else:
    return (string[:2]+string[-2:])

print(join_f_l_2(string))
```

```
Thct
```

4. Write a Python program to get a string from a given string where all occurrences of its first char have been changed to '$', except the first char itself.

```
def replace_fchar_with_dollar(inputs):
  string1 = string.replace(inputs[0], '$')
  return (inputs[0] + inputs[1:])

inputs = input("Enter the string to count characters >>>> ")
print(replace_fchar_with_dollar(inputs))
```

```
Enter the string to count characters >>>> rebecca dias
rebecca dias
```

5. Write a Python program to get a single string from two given strings, separated by a space and swap the first two characters of each string.

```
def swap_first_two(string1,string2):
  str1 = string2[:2] + string1[2:]
  str2 = string1[:2] + string2[2:]
  return (str1+" "+str2)

string1, string2 = input("Enter the two string separated by space for swap >>>> ").split(" ")
print(swap_first_two(string1, string2))
```

```
Enter the two string separated by space for swap >>>> reb dias
dib reas
```

6. Write a Python program to add 'ing' at the end of a given string (length should be at least 3). If the given string already ends with 'ing' then add 'ly' instead. If the string length of the given string is less than 3, leave it unchanged.

```
def add_ing(string):
  if len(string) > 2:
    if string[-3:] == 'ing':
      string += 'ly'
    else:
      string += 'ing'
  return string

string = input("Enter the string >>>> ")
print(add_ing(string))
```

```
Enter the string >>>> rebly
reblying
```

7. Write a Python program to find the first appearance of the substring 'not' and 'poor' from a given string, if 'not' follows the 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.

```
def not_poor_edit(string):
  nott = string.find('not')
  poor = string.find('poor')
  if (poor > nott and nott > 0 and poor > 0):
    string = string.replace(string[nott:(poor+4)], 'good')
    return string
  else:
    return string
string = input("Enter the string >>>> ")
print(not_poor_edit(string))
```

```
Enter the string >>>> i am not poor
```

```
i am good
```

8. Write a Python function that takes a list of words and returns the length of the longest one.

```
def find_longest(str_list):
  str_list_l = []
  for i in str_list:
    str_list_l.append((len(i), i))
  str_list_l.sort()

  return (str_list_l[-1][0], str_list_l[-1][1])

str_list = list(input("Enter the words space separated >>>> ").split(" "))
print(find_longest(str_list))
```

```
Enter the words space separated >>>> rebecca dias
(7, 'rebecca')
```

9. Write a Python program to remove the nth index character from a nonempty string.

```
def remove_n(string, n):
  f = string[:n]
  l = string[n+1:]
  return f + l
string, n = input("Enter the string and the nth index separated by a space >>>> ").split(" ")
print(remove_n(string, int(n)))
```

```
Enter the string and the nth index separated by a space >>>> rebecca 3
rebcca
```

10. Write a Python program to change a given string to a new string where the first and last chars have been exchanged.

```
def replace_f_l(string):
  return string[-1:] + string[1:-1] + string[:1]
string = input("Enter the string >>>> ")
print(replace_f_l(string))
```

```
Enter the string >>>> rebecca dias
sebecca diar
```

11. Write a Python program to remove the characters which have odd index values of a given string.

```
def remove_odd_index(string):
  return (string[::2])
string = input("Enter the string >>>> ")
print(remove_odd_index(string))
```

```
Enter the string >>>> sachin
sci
```

12. Write a Python program to count the occurrences of each word in a given sentence.

```
def count_occr(string):
  word_c = {}
  words = string.split()
  for i in words:
    if i in word_c:
      word_c[i] += 1
    else:
```

```
      word_c[i] = 1
  return word_c
string = input("Enter the string >>>> ")
print(count_occr(string))
```

```
Enter the string >>>> i am a i am good boy
{'i': 2, 'am': 2, 'a': 1, 'good': 1, 'boy': 1}
```

13. Write a Python script that takes input from the user and displays that input back in upper and lower cases.

```
def get_up_low(string):
  return(string.upper()+" "+string.lower())
string = input("Enter the string >>>> ")
print(get_up_low(string))
```

```
Enter the string >>>> hey
HEY hey
```

14. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically).

```
def get_unique(string):
  words = [word for word in string.split(",")]
  return (",".join(sorted(list(set(words)))))
string = input("Enter the string >>>> ")
print(get_unique(string))
```

```
Enter the string >>>> hello,world,i,am,reb
am,hello,i,reb,world
```

15.Write a Python function to create the HTML string with tags around the word(s).

```
def html_tag(tag, word):
    return (f"<{tag}> {word} </{tag}>")
word, tag = input("Enter the word and tag space separated >>>> ").split(" ")
print(html_tag(tag, word))
```

```
Enter the word and tag space separated >>>> hello h2
<h2> hello </h2>
```

Conclusion:

Implemented various string processing techniques on strings in Python.We used the Built-in Functions in string and performed various programming exercises.