Strings

August 12, 2023

[ ]:
```python
# Write a python program to calculate a string
A = "Pratham"
B = len(A)
print(B)


# OR
def len(str):
    count = 0
    for char in str:
        count += 1
    return count


print(len("pratham"))
```

```
7
7
```

[ ]:
```python
# write a program to count the number of characters(character frequency) in a string
def char_freq(A):
    dict = {}
    for s in A:
        keys = dict.keys()
        if s in keys:
            dict[s] += 1
        else:
            dict[s] = 1
    return dict


char_freq("Student Management")
```

[ ]:
```
{'S': 1,
 't': 3,
 'u': 1,
 'd': 1,
```

```
 'e': 3,
 'n': 3,
 ' ': 1,
 'M': 1,
 'a': 2,
 'g': 1,
 'm': 1}
```

[ ]:
```
# Write a Python program to get a string made of the first 2 and last 2␣
 ↪characters of a given string.
# If the string length is less than 2, return the empty string instead.
def string(C):
    if len(C) < 2:
        print("empty string")
    else:
        print(C[0:2] + C[-2:])


C = input("enter a string")

print(string(C))
```

```
pram
None
```

[ ]:
```
# Write a Python program to get a string from a given string where all␣
 ↪occurrences of its first char have been changed to '$', except the first␣
 ↪char itself.
def string(M):
    char = M[0]
    M = M.replace(char, "$")
    M = char + M[1:]
    print(M)


A = string(input())
```

```
apoorv$
```

[ ]:
```
# 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.
str1 = input()
str2 = input()
print(str1)
print(str2)

print(str2 + " " + str1)
```

```
sss
```

```
yyy
yyy sss
```

[ ]:
```python
# 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', add 'ly' instead.
# If the string length of the given string is less than 3, leave it unchanged.
def string():
    str = input("Enter your word: ")
    if len(str) >= 3 and str[-3:] == "ing":
        return str + "ly"
    else:
        return str + "ing"


string()
```

[ ]:
```
'readingly'
```

[ ]:
```python
# Write a Python program to find the first appearance of the substrings 'not' and 'poor' in a given string.
# If 'not' follows 'poor', replace the whole 'not'...'poor' substring with 'good'. Return the resulting string.
def not_poor(str1):
    snot = str1.find("not")
    spoor = str1.find("poor")

    if spoor > snot and snot > 0 and spoor > 0:
        str1 = str1.replace(str1[snot : (spoor + 4)], "good")
        return str1
    else:
        return str1


print(not_poor("The lyrics is not that poor!"))
print(not_poor("The lyrics is poor!"))
```

```
The lyrics is good!
The lyrics is poor!
```

[ ]:
```python
# Write a Python function that takes a list of words and return the longest word and the length of the longest one.


def find_longest(words):
    longest = ""
    for word in words:
        if len(word) > len(longest):
```

```
            longest = word
    return longest


input_str = input("Enter a list of words separated by spaces: ")
print(input_str)
word_list = input_str.split()
result = find_longest(word_list)
print("Longest word:", result)
```

```
an apple a day
Longest word: apple
```

[ ]:
```
# Write a Python program to change a given string to a newly string where the first and last chars have been exchanged.

str = input()
print(str)

print("The newly string is", str[-1] + str[1:-1] + str[0])
```

```
doctor
The newly string is roctod
```

[ ]:
```
# Write a Python program that accepts a comma-separated sequence of words as input and prints the distinct words in sorted form (alphanumerically).
string = input("Enter a comma seperated string")
a = string.split(",")
sorted(a)
```

[ ]:
```
['justin pratham punit tanga ']
```

[ ]:
```
string = input("Enter a space seperated string")
str1 = input("enter string to be added")
B = print(string + " " + str1)
```

```
pratham justin punit tanga vinay johanan kushal anush
```

[ ]:
```
# insert string in middle
def insert_sting_middle(str, word):
    return str[:2] + word + str[2:]


print(insert_sting_middle("[[]]", "Python"))
print(insert_sting_middle("{{}}", "PHP"))
print(insert_sting_middle("<<>>", "HTML"))
```

```
[[Python]]
{{PHP}}
```

<<HTML>>

[ ]: