Programs on Immutable Data Structure:

In [79]:
```
#1. write a python program to find a length of a string without using len function
string=input("Enter string:")
count=0
for i in string:
      count=count+1
print("Length of the string is:")
print(count)
```

```
Enter string:Python
Length of the string is:
6
```

In [80]:
```
#2. Write a Python Program to check if a string is Palindrome or not using function.
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Enter string:"))
if(is_palindrome(a)==True):
    print("String is a palindrome!")
else:
    print("String isn't a palindrome!")
```

```
Enter string:Python
String isn't a palindrome!
```

In [81]:
```
#2. Write a Python Program to check if a string is Palindrome or not using function.
def is_palindrome(s):
    if len(s) < 1:
        return True
    else:
        if s[0] == s[-1]:
            return is_palindrome(s[1:-1])
        else:
            return False
a=str(input("Enter string:"))
if(is_palindrome(a)==True):
    print("String is a palindrome!")
else:
    print("String isn't a palindrome!")
```

```
Enter string:mom
String is a palindrome!
```

In [1]:
```
#3. Create a string made of the first, middle and last character
str1 = 'James'
print("Original String is", str1)
# Get first character
res = str1[0]
# Get string size
```

```
l = len(str1)
# Get middle index number
mi = int(l / 2)
# Get middle character and add it to result
res = res + str1[mi]
# Get last character and add it to result
res = res + str1[l - 1]
print("New String:", res)
```

```
Original String is James
New String: Jms
```

In [4]:
```
#4. Python Program to Create a New String Made up of First and Last 2 Characters
string=input("Enter string:")
count=0
for i in string:
      count=count+1
new=string[0:2]+string[count-2:count]
print("Newly formed string is:")
print(new)
```

```
Enter string:Python is very important
Newly formed string is:
Pynt
```

In [6]:
```
#5. Python Program to Find the Larger String without using Built-in Functions
string1=input("Enter first string:")
string2=input("Enter second string:")
count1=0
count2=0
for i in string1:
      count1=count1+1
for j in string2:
      count2=count2+1
if(count1<count2):
      print("Larger string is:")
      print(string2)
elif(count1==count2):
      print("Both strings are equal.")
else:
      print("Larger string is:")
      print(string1)
```

```
Enter first string:Python is very logicable
Enter second string:Python is very interesting to learn
Larger string is:
Python is very interesting to learn
```

In [11]:
```
#6. Count all letters, digits, and special symbols from a given string.
def find_digits_chars_symbols(sample_str):
    char_count = 0
    digit_count = 0
    symbol_count = 0
    for char in sample_str:
        if char.isalpha():
            char_count += 1
        elif char.isdigit():
            digit_count += 1
```

```
# if it is not letter or digit then it is special symbol
        else:
            symbol_count += 1
    print("Chars =",char_count, "Digits =", digit_count, "Symbol =", symbol_count)
sample_str = "P@yn2at&#i5ve"
print("total counts of chars, Digits, and symbols \n")
find_digits_chars_symbols(sample_str)
```

```
total counts of chars, Digits, and symbols

Chars = 8 Digits = 2 Symbol = 3
```

In [13]:
```
#7. Write a Python function that accepts a string and calculate the number of uppercas
n=input("Enter String ")
upper_counter=0
lower_counter=0
for x in n:
    if x.isupper():
        upper_counter+=1
    elif x.islower():
        lower_counter+=1
    else:
        pass
print("The number of uppercase letters is ",upper_counter)
print("The number of lowercase letters is ",lower_counter)
```

```
Enter String Python is a very Logical Subject
The number of uppercase letters is  3
The number of lowercase letters is  24
```

In [14]:
```
#8. Find all occurrences of a substring in a given string by ignoring the case.
str1 = "Welcome to USA. usa awesome, isn't it?"
sub_string = "USA"
# convert string to lowercase
temp_str = str1.lower()
# use count function
count = temp_str.count(sub_string.lower())
print("The USA count is:", count)
```

```
The USA count is: 2
```

In [15]:
```
#9. Calculate the sum and average of the digits present in a string.
input_str = "PYnative29@#8496"
total = 0
cnt = 0
for char in input_str:
    if char.isdigit():
        total += int(char)
        cnt += 1
# average = sum / count of digits
avg = total / cnt
print("Sum is:", total, "Average is ", avg)
```

```
Sum is: 38 Average is  6.333333333333333
```

In [16]:
```
#10. Write a program to Reverse a given string
str1 = "PYnative"
print("Original String is:", str1)
```

```
str1 = str1[::-1]
print("Reversed String is:", str1)
```

```
Original String is: PYnative
Reversed String is: evitanYP
```

In [17]:
```
#11. Split a string on hyphens.
#Given:
#str1 = Emma-is-a-data-scientist
#Expected Output:
#Displaying each substring
#Emma
#is
#a
#data
#scientist
str1 = "Emma-is-a-data-scientist"
print("Original String is:", str1)
# split string

sub_strings = str1.split("-")
print("Displaying each substring")
for sub in sub_strings:
    print(sub)
```

```
Original String is: Emma-is-a-data-scientist
Displaying each substring
Emma
is
a
data
scientist
```

In [18]:
```
#12. Write a program to do sum of tuple elements.
test_tup = (1, 2, 3)
sum=0
for i in test_tup:
    sum=sum+i
print(sum)
```

```
6
```

In [20]:
```
#13. Write a program to print Maximum and Minimum elements in given Tuple
test_tup = (1, 2, 3)
max=test_tup[0]
min=test_tup[0]
for i in test_tup:
    if (i>max):
        max=i
    if (i<min):
        min=i
print(max)
print(min)
```

```
3
1
```

In [21]:
```
#14. Write a program to print even numbers from given tuple.
test_tup = (1, 2, 3, 4, 5, 6)
```

```
for i in test_tup:
    if (i%2==0):
        print(i)
```

```
2
4
6
```

In [22]:
```
#15. Write a program to print sum of even numbers and sum of odd numbers from elements
test_tup = (1, 2, 3, 4, 5, 6)
odd=0
even=0
for i in test_tup:
    if (i%2==0):
        even+=i;
    else:
        odd+=i
print("Odd sum: ",odd)
print("Even sum: ",even)
```

```
Odd sum:  9
Even sum:  12
```

In [23]:
```
#16. Python Program to Check if a Date is Valid and Print the Incremented Date if it i
date=input("Enter the date: ")
dd,mm,yy=date.split('/')
dd=int(dd)
mm=int(mm)
yy=int(yy)
if(mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or mm==10 or mm==12):
    max1=31
elif(mm==4 or mm==6 or mm==9 or mm==11):
    max1=30
elif(yy%4==0 and yy%100!=0 or yy%400==0):
    max1=29
else:
    max1=28
if(mm<1 or mm>12):
    print("Date is invalid.")
elif(dd<1 or dd>max1):
    print("Date is invalid.")
elif(dd==max1 and mm!=12):
    dd=1
    mm=mm+1
    print("The incremented date is: ",dd,mm,yy)
elif(dd==31 and mm==12):
    dd=1
    mm=1
    yy=yy+1
    print("The incremented date is: ",dd,mm,yy)
else:
    dd=dd+1
    print("The incremented date is: ",dd,mm,yy)
```

```
Enter the date: 17/10/22
The incremented date is:  18 10 22
```

In [24]:
```
#16. Python Program to Check if a Date is Valid and Print the Incremented Date if it i
date=input("Enter the date: ")
dd,mm,yy=date.split('/')
dd=int(dd)
mm=int(mm)
yy=int(yy)
if(mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or mm==10 or mm==12):
    max1=31
elif(mm==4 or mm==6 or mm==9 or mm==11):
    max1=30
elif(yy%4==0 and yy%100!=0 or yy%400==0):
    max1=29
else:
    max1=28
if(mm<1 or mm>12):
    print("Date is invalid.")
elif(dd<1 or dd>max1):
    print("Date is invalid.")
elif(dd==max1 and mm!=12):
    dd=1
    mm=mm+1
    print("The incremented date is: ",dd,mm,yy)
elif(dd==31 and mm==12):
    dd=1
    mm=1
    yy=yy+1
    print("The incremented date is: ",dd,mm,yy)
else:
    dd=dd+1
    print("The incremented date is: ",dd,mm,yy)
```

```
Enter the date: 5/7/2004
The incremented date is:  6 7 2004
```

In [25]:
```
#16. Python Program to Check if a Date is Valid and Print the Incremented Date if it i
date=input("Enter the date: ")
dd,mm,yy=date.split('/')
dd=int(dd)
mm=int(mm)
yy=int(yy)
if(mm==1 or mm==3 or mm==5 or mm==7 or mm==8 or mm==10 or mm==12):
    max1=31
elif(mm==4 or mm==6 or mm==9 or mm==11):
    max1=30
elif(yy%4==0 and yy%100!=0 or yy%400==0):
    max1=29
else:
    max1=28
if(mm<1 or mm>12):
    print("Date is invalid.")
elif(dd<1 or dd>max1):
    print("Date is invalid.")
elif(dd==max1 and mm!=12):
    dd=1
    mm=mm+1
    print("The incremented date is: ",dd,mm,yy)
elif(dd==31 and mm==12):
```

```
    dd=1
    mm=1
    yy=yy+1
    print("The incremented date is: ",dd,mm,yy)
else:
    dd=dd+1
    print("The incremented date is: ",dd,mm,yy)
```

```
Enter the date: 30/2/1997
Date is invalid.
```

In [27]:
```
#17. Python Program to Find the Smallest Divisor of an Integer

n=int(input("Enter an integer:"))
a=[]
for i in range(2,n+1):
    if(n%i==0):
        a.append(i)
a.sort()
print("Smallest divisor is:",a[0])
```

```
Enter an integer:45
Smallest divisor is: 3
```

In [1]:
```
"""18. Write a Python program that takes a string and encode it that the amount of sym
For example, the string "AAAABBBCCDAAA" would be encoded as "4A3B2C1D3A"""
def encode_string(str1):
    encoded = ""
    ctr = 1
    last_char = str1[0]

    for i in range(1, len(str1)):

        if last_char == str1[i]:
            ctr += 1

        else:
            encoded += str(ctr) + last_char
            ctr = 0
            last_char = str1[i]
            ctr += 1
    encoded += str(ctr) + last_char
    return encoded
print(encode_string("AAAABBBCCDAAA"))
print(encode_string("PHP"))
print(encode_string("AAAABBBCCCDAABDAAAAC"))
```

```
4A3B2C1D3A
1P1H1P
4A3B3C1D2A1B1D4A1C
```

In [2]:
```
"""19. From Wikipedia, the free encyclopedia
In computational linguistics and computer science, edit distance is a way of quantifyi
This problem was asked by Google.
The edit distance between two strings refers to the minimum number of character insert
Write a Python program to compute the edit distance between two given strings."""
def edit_distance(string1, string2):
```

```
    if len(string1) > len(string2):
        difference = len(string1) - len(string2)
        string1[:difference]

    elif len(string2) > len(string1):
        difference = len(string2) - len(string1)
        string2[:difference]

    else:
        difference = 0

    for i in range(len(string1)):
        if string1[i] != string2[i]:
            difference += 1

    return difference

print(edit_distance("kitten", "sitting")) #3
print(edit_distance("medium", "median")) #2
```

```
3
2
```

In [ ]: