Q 1 .Write a python program to sum of the first n positive integers.

In [2]:

```python
N = int(input("Enter a number"))

SUM = 0
for num in range(1, N+1) :
    SUM += num

print(f"Sum of {N} numbers is:  {SUM}")

```

```
Enter a number4
Sum of 4 numbers is:  10
```

Q2. Write a Python program to count occurrences of a substring in a string.

In [16]:

```python
S = input("Enter a String:  ")
sub_s = input("Enter a Substring:  ")
Leng_S = len(S)
print("\n")
print(f"Length of the String is : {Leng_S}")
print("\n")
frequency  =S.count(sub_s)
print(f" The frequency of the {sub_s} in the {S} is : {frequency} ")

```

```
Enter a String:  aabbcc nnmmoopp gghhaabbcc
Enter a Substring:  bbcc


Length of the String is : 26


 The frequency of the bbcc in the aabbcc nnmmoopp gghhaabbcc is : 2
```

Q3.Write a Python program to count the occurrences of each word in a given sentence

In [46]:

```python
count = 0
start = 0
String = input("Enter a string with more than 2 strings :   ")
substring =  input("Enter a string :   ")

L = String.split()
print(L[1:3])
length_string = len(L)
print("length of string : ",length_string )
length_substring = len(substring)


for i in range(0,length_string ):
    if L[i] == substring:
        count +=1

    else:
        Flag ="Red"

print("count is ", count)
```

```
Enter a string with more than 2 strings :   gayatri Dv
Enter a string :   Dv
['Dv']
length of string :  2
count is  1
```

Q4.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

In [7]:

```
s1 = input("Enter a string:  ")
s2 = input("Enter another string:  ")

s1[0]

print("\n Swapped_s1_chars : ", s1[0]+s1[1])

print("\n Swapped_s2_chars : ", s2[0]+s2[1])

Modified_string1 =  s2[0]+s2[1] +s1[2:]
Modified_string2 = s1[0]+s1[1] + s2[2:]



print("Swapped_s1 = ", Modified_string1)

print("Swapped_s2 = ", Modified_string2)
```

```
Enter a string:  abcdef
Enter another string:  ghijkl

 Swapped_s1_chars :  ab

 Swapped_s2_chars :  gh
Swapped_s1 =  ghcdef
Swapped_s2 =  abijkl
```

Q5.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

In [22]:

```
S = input("Enter a string :")
Length_S = len(S)

print("Length of the String : ",Length_S)


if Length_S < 3:
    print(" length is less than 3")
elif (Length_S >= 3) and (S[-3:] == "ing"):
    New_string = S+ "ly"

elif (Length_S >= 3) and (S[-3:] != "ing"):
    New_string = S+ "ing"

print("Modified String is = ",New_string)


```

```
Enter a string :sdfghjing
Length of the String :  9
sdfghjingly
```

Q6.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

In [12]:

```
S = input("Enter a string: ")
Length_S = len(S)
not_number =  " "
poor_number = " "
Result = " "
Str= " "

not_number = int(S.find("not"))
poor_number = int(S.find("poor"))

print("not_number =" , not_number)
print("poor_number =" , poor_number)

if poor_number > not_number:
     #S.(S[poor_number:not_number+3] , "good")
        Str =S[not_number:poor_number+4 ]
        Result = S.replace(Str,"good")


print("Replaced value: ", Str)
print("Final String =",Result)

```

```
Enter a string: aaaa_not_bbb_poor_cccc
not_number = 5
poor_number = 13
Replaced value:  not_bbb_poor
Final String = aaaa_good_cccc
```

In [2]:

```
S = input("Enter a string: ")
words = S.split()
count ={}

for word in words:
    if word in count:
        count[word]+=1
    else:
        count[word]= 1

print("Word Count : ", count)



```

```
Enter a string: gayatri dinavahi gayatri
Word Count :  {'gayatri': 2, 'dinavahi': 1}
```

In [11]:

```python
S = input("Enter a string: ")
List_words = S.split()
print(List_words)
Count_Words ={}
for words in List_words:
    if words in Count_Words :
        Count_Words[words]+= 1
    else :
        Count_Words[words] =1
print(count )
```

```
Enter a string: gayatri dinavahi gayatri
['gayatri', 'dinavahi', 'gayatri']
1
```

Q7.Program to find Greatest Common Divisor of two numbers. For example, the GCD of 20 and 28 is 4 and GCD of 98 and 56 is 14.

In [39]:

```python
N1 = int(input("Enter an integer number N1 : "))
N2 = int(input("Enter an integer number N2 : "))
L = []



def GCD_funt(A,B):
    GCD = 1
    if B> A:
        greater_num = B
        Smaller_num = A
        for factor in range (1,Smaller_num+1 ):
            if (A% factor == 0) and (B% factor ==0):
                L.append(factor)
    elif A>B:
        greater_num = A
        Smaller_num = B
        for factor in range (1,Smaller_num+1 ):
            if (B% factor == 0) and (A% factor ==0):
                L.append(factor)
    elif A==B:
        L.append(A)
    GCD = max(L)
    return GCD

output = GCD_funt(N1,N2)

print(f"\nGCD of {N1} and {N2} is {output} ")

```

```
Enter an integer number N1 : 108
Enter an integer number N2 : 96

GCD of 108 and 96 is 12
```

Q8.Write a Python program to check whether a list contains a sublist.

In [6]:

```python
List1 = [2,3,[4,5,6]]
sublist = [4,5,6]

Length_List1 = len(List1)
print("Length_List  =", Length_List1)
Length_sublist = len(sublist)
print("Length_sublist =", Length_sublist)

Flag = "Red"

for element in range(0 , Length_List1 ):
    if List1[element] == sublist:
        Flag = "Green"
        break
    else:
        print("_")

if Flag == "Green" :
    print(f" {sublist} is one of the elements in List1 - {List1}")
else:
    print("sorry")

```

```
Length_List  = 3
Length_sublist = 3
_
_
 [4, 5, 6] is one of the elements in List1 - [2, 3, [4, 5, 6]]
```

Q9.Write a Python program to find the second smallest number in a list.

In [14]:

```python
# throught sort function

L = [3,4,5,60,7,2,1,-4,-28,-98,-150, 10,100]

L.sort()

print("Sorted List : " , L)

print("Second Smallest element in the List :  " , L[1])
```

```
Sorted List :  [-150, -98, -28, -4, 1, 2, 3, 4, 5, 7, 10, 60, 100]
Second Smallest element in the List :   -98
```

In [31]:

```python
L1 =  [3,4,5,60,7,2,1,-4,-28,-98,-150, 10,100]

print(L1)

Smallest =min(L1)
Smallest_index = L1.index(Smallest)

print("Smallest =" , Smallest)
print("Smallest_index = ", Smallest_index)
L1.pop(Smallest_index)
print(L1)

Second_Smallest = min(L1)

print("Second_Smallest =" ,Second_Smallest)
```

```
[3, 4, 5, 60, 7, 2, 1, -4, -28, -98, -150, 10, 100]
Smallest = -150
Smallest_index =  10
[3, 4, 5, 60, 7, 2, 1, -4, -28, -98, 10, 100]
Second_Smallest = -98
```

Q10. Write a Python program to get unique values from a list.

In [32]:

```python
# through set() Function
L2 = [3,4,5,3,4,5,1,2,6,7,8,6,7,8]
unique_list = []

unique_list = set(L2)

print(unique_list)
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

In [42]:

```python
#  through Logic
L2 =[2,3,4,2,3]

#[3,4,5,3,4,5,1,2,6,7,8,6,7,8]
unique_list = []
length_L2 = len(L2)
print("Length = ",length_L2)

for item in L2:
    if item not in unique_list:
        unique_list.append(item)
print("unique_list =" , unique_list)
```

```
Length =  5
unique_list = [2, 3, 4]
```

Q11. Write a Python program to unzip a list of tuples into individual lists.

In [17]:

```
L = [("a", 1) , ("b",2),("c",3),("d",4)]

Result = list(zip(*L))

print("Unzipped list of tupples is ",Result)
```

```
Unzipped list of tupples is  [('a', 'b', 'c', 'd'), (1, 2, 3, 4)]
```

Q12. Write a Python program to convert a list of tuples into a dictionary

In [ ]:

```
# Tuple1= [("Aditya" , 1),("Bhuvan", 2),("Chintan", 3),("David", 4)]
Dict = {}
def Tupple_to_Dictionery(T,D) :
    for Key , Value in T :
        D.setdefault(Key,[]).append(Value)
    return D

Result = Tupple_to_Dictionery(Tuple1 , Dict)

print(Result)
```

Q13. Write a Python program to sort a dictionary (ascending /descending) by value.

In [105]:

```python
D = {"Aditya" : 1111 ,
     "Shreekant" :2222,
     "Raman" : 3333,
     "TaraK" : 5555,
     "Udit":7777,
     "Manan": 8888,
     "Nishant" :9999,
     "Aman": 6666
    }

D_string = str(D)
print("Original Dictionery :  ",D_string)

# sorting by the Keys
Sorted_by_Keys = dict(sorted(D.items()))
print("\nSorted by keys: ",Sorted_by_Keys )
type(Sorted_by_Keys)

# sorting by the values:
result = {key:val for key, val in sorted(D.items(), key = lambda value: va
print("sorting by the values: ",result)

```

```
Original Dictionery :   {'Aditya': 1111, 'Shreekant': 2222, 'Raman': 3333, 'T
araK': 5555, 'Udit': 7777, 'Manan': 8888, 'Nishant': 9999, 'Aman': 6666}

Sorted by keys:  {'Aditya': 1111, 'Aman': 6666, 'Manan': 8888, 'Nishant': 999
9, 'Raman': 3333, 'Shreekant': 2222, 'TaraK': 5555, 'Udit': 7777}
sorting by the values:  {'Aditya': 1111, 'Shreekant': 2222, 'Raman': 3333, 'T
araK': 5555, 'Aman': 6666, 'Udit': 7777, 'Manan': 8888, 'Nishant': 9999}
```

In [67]:

```python
Numbers= [1,2,2,4,4,4,4,3,3,3,7,7,7,5,5,5]
N =sorted(Numbers)
print(N)
```

```
[1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 7, 7, 7]
```

Q14.Write a Python program to find the highest 3 values in a dictionary.

In [86]:

```python
D = {"Aman" : 3333,
     "Gayatri":2222,
     "Akash" : 6666,
     "Manan" : 8888,
     "Nitin": 9999
    }
Highest_3_values =sorted(D.values(), reverse= True)
print("Highest 3 values in the Dictionery are:",Highest_3_values[:3])



```

```
[9999, 8888, 6666]
```

Q15. Given a number n, write a python program to make and print the list of Fibonacci series up to n.

Input : n=7 Hint : first 7 numbers in the series Expected output : First few Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13

In [5]:

```python
N = int(input("Enter a number :"))

Fib = []

Fib.append(0)
Fib.append(1)

for i in range (2, N):
    Num = Fib[i-1]+ Fib[i-2]
    Fib.append(Num)

print(Fib)
```

```
Enter a number :5
[0, 1, 1, 2, 3]
```

Q16.Counting the frequencies in a list using a dictionary in Python.

Input : [1, 1, 1, 5, 5, 3, 1, 3, 3, 1,4, 4, 4, 2, 2, 2, 2]

Expected output : 1 : 5 , 2 : 4 , 3 : 3 , 4 : 3 , 5 : 2

In [13]:

```
Original_List = [1, 1, 1, 5, 5, 3, 1, 3, 3, 1,4, 4, 4, 2, 2, 2, 2]

def count_frequency_elements(List):

    frequencies = {}
    for item in Original_List:
        if item in frequencies:
            frequencies[item] +=1
        else:
            frequencies[item] =1


    for key, value in frequencies.items():
        print(" %d : %d " % (key, value))
    return(frequencies)

result_Dict = count_frequency_elements(Original_List)
print("result_Dict : ", result_Dict)

```

```
 1 : 5 
 5 : 2 
 3 : 3 
 4 : 3 
 2 : 4 
result_Dict :  {1: 5, 5: 2, 3: 3, 4: 3, 2: 4}
```

In [24]:

```python
L = [1,2,3,4,1,1,3,4,5,2,2,4,5,6,3,3,4,5,7,8,9,7,8,9]
Length_L = len(L)
SET = set(L)
print("SET = ", SET)
print("L = ", L)
unique = []
freq_dict ={}

for element in L:
    print(L[element], end= " ")
    if L[element] not in unique:
        unique.append(L[element])

print("\nunique =" , unique)
for item in L :
    if item in freq_dict:
        freq_dict[item] += 1
    else:
        freq_dict[item] =1
print("Frequency of the elements is as follows: \n" ,freq_dict)
print("\n")
for K, V in freq_dict.items():
    print("%d : %d" % (K, V))
```

```
SET =  {1, 2, 3, 4, 5, 6, 7, 8, 9}
L =  [1, 2, 3, 4, 1, 1, 3, 4, 5, 2, 2, 4, 5, 6, 3, 3, 4, 5, 7, 8, 9, 7, 8, 9]
2 3 4 1 2 2 4 1 1 3 3 1 1 3 4 4 1 1 4 5 2 4 5 2
unique = [2, 3, 4, 1, 5]
Frequency of the elements is as follows: 
 {1: 3, 2: 3, 3: 4, 4: 4, 5: 3, 6: 1, 7: 2, 8: 2, 9: 2}


1 : 3
2 : 3
3 : 4
4 : 4
5 : 3
6 : 1
7 : 2
8 : 2
9 : 2
```

Q17. Write a python program using function to find the sum of odd series and even series

Odd series: 12/ 1! + 32/ 3! + 52/ 5!+……n Even series: 22/ 2! + 42/ 4! + 62/ 6!+……n

In [50]:

```
#Odd Series = 1!  + 3!+ 5!.....n
#Even Series = 2!+4! +6!.......n


Num = int(input("Enter a number of your choice:  "))


def Fact_summation(N):
    i_even = 2
    i_odd = 1
    Result_Sum =0
    Status = "Odd"

    fact= 1
    if (N%2) ==0 :
        Status = "Even"
    elif (N%2) !=0:
        Status = "Odd"

    if Status == "Even":
        while i_even <= N:
            fact*= iteration
            i_even +=2
        Result_Sum += fact

    else:
        while i_odd <=N:
            fact*= iteration
            i_odd +=2
        Result_Sum += fact

    return Result_Sum

Output =  Fact_summation(Num)

print("Output :  ", Output)

```

```
Enter a number of your choice:  3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[50], line 34
     30         Result_Sum += fact
     32     return Result_Sum
---> 34 Output =  Fact_summation(Num)
     36 print("Output :  ", Output)

Cell In[50], line 28, in Fact_summation(N)
     26 else:
     27     while i_odd <=N:
---> 28         fact*= iteration
     29         i_odd +=2
     30     Result_Sum += fact

NameError: name 'iteration' is not defined
```

Q18. Python Program to Find Factorial of Number Using Recursion

In [5]:

```python
N = int(input("Enter an integer number for which factorial is to be calcul
Fact = 1
i = 1
SUM = 0
while i <=N:

    Fact= Fact * i
     i=i+1
    print("i= ", i)
    print("\tFact = " , Fact)

print("\n")
print(f"Result of the Factorial of {N } is : {Fact} " )

```

```
Enter an integer number for which factorial is to be calculated :  3
i=  2
        Fact =  1
i=  3
        Fact =  2
i=  4
        Fact =  6




Result of the Factorial of 3 is : 6
```

Q19. Write a Python function that takes a list and returns a new list with unique elements of

In [36]:

```python
L = [1,1,2,3,3,4,8,9,2,6,5,9,4,4,5,6,7,8,2,3,4,5,1,2,3,4,6,9,1,2,4,6,7,3,
print(" L : ", L)


# through Logig
unique = []

for item in  L:
    if item not in unique:
        unique.append(item)

print("unique : ", unique)
Sorted_unique = sorted(unique)

# through SET() Function in Python
SET_unique = set(L)
print("SET =  ", SET )

```

```
 L :  [1, 1, 2, 3, 3, 4, 8, 9, 2, 6, 5, 9, 4, 4, 5, 6, 7, 8, 2, 3, 4, 5, 1,
2, 3, 4, 6, 9, 1, 2, 4, 6, 7, 3, 7, 8]
Length of L :  36
unique :  [1, 2, 3, 4, 8, 9, 6, 5, 7]
SET =   {1, 2, 3, 4, 5, 6, 7, 8, 9}
```

In [ ]:

In [ ]:

In [ ]: