In [ ]:
```
# Function Decorators:

# Decorator is a function which can take a function as argument and extend its func
# and returns modified function with extended functionality.
# Explain decorator with real  life example:-

# In below example in main function numerator 2 and denominator is 4, so  output is
# So,without change/ touching the main function swap the numerator and denominator

def Smart_div(func):
    def inner(a,b):
        if a<b:
            a,b=b,a
        return func(a,b)

    return inner


@Smart_div
def div(a,b):
    print(a/b)


div(2,4)
```

In [ ]:
```
# Generators:-Generator is a function which is responsible to generate a sequence o

# We can write generator functions just like ordinary functions, but it uses yield



def simple():
    print('Even Number')
    for i in range(10):
        if(i%2==0):
            yield i
x=simple()
for i in x:
    print(i)


# yield vs. return :-

    # The yield statement is responsible for controlling the flow of the generator
    # The return statement returns a value and terminates the whole function an onl
       # in the function.



# Difference between Generator function and Normal function:-

# Normal function contains only one return statement whereas generator function can

# When the generator functions are called, the normal function is paused immediatel

# The return statement returns a value and terminates the whole function.




# Generator Expression:-
# The representation of generator expression is similar to the Python list comprehe
  # The only difference is that square bracket is replaced by round parentheses.
    # The list comprehension calculates the entire list, whereas the generator expr

list = [1,2,3,4,5,6,7]

# List Comprehension
z = [x**3 for x in list]
print(z)

# Generator expression
a = (x**3 for x in list)
print(next(a))
print(next(a))
```

In [ ]:

```
# Date-Time Programs:



# Format        Description
# %a    Abbreviated weekday name (Sun to Sat)
# %b    Abbreviated month name (Jan to Dec)
# %c    Numeric month name (0 to 12)
# %D    Day of the month as a numeric value, followed by suffix (1st, 2nd, 3rd, ..
# %d    Day of the month as a numeric value (01 to 31)
# %e    Day of the month as a numeric value (0 to 31)length
# %f    Microseconds (000000 to 999999)
# %H    Hour (00 to 23)
# %h    Hour (00 to 12)
# %I    Hour (00 to 12)
# %i    Minutes (00 to 59)
# %j    Day of the year (001 to 366)
# %k    Hour (0 to 23)a
# %l    Hour (1 to 12)
# %M    Month name in full (January to December)
# %m    Month name as a numeric value (00 to 12)
# %p    AM or PM
# %r    Time in 12 hour AM or PM format (hh:mm:ss AM/PM)
# %S    Seconds (00 to 59)
# %s    Seconds (00 to 59)
# %T    Time in 24 hour format (hh:mm:ss)
# %U    Week where Sunday is the first day of the week (00 to 53)
# %u    Week where Monday is the first day of the week (00 to 53)
# %V    Week where Sunday is the first day of the week (01 to 53). Used with %X
# %v    Week where Monday is the first day of the week (01 to 53). Used with %x
# %W    Weekday name in full (Sunday to Saturday)
# %w    Day of the week where Sunday=0 and Saturday=6
# %X    Year for the week where Sunday is the first day of the week. Used with %V
# %x    Year for the week where Monday is the first day of the week. Used with %v
# %Y    Year as a numeric, 4-digit value
# %y    Year as a numeric, 2-digit value





print(dir(datetime))
```

In [1]:

```
# 1. Python program to get Current Time?

from datetime import datetime

CDT= datetime.now().date()
print(CDT)
```

```
CT=datetime.now().time()
print(CT)

CT_str12=CT.strftime("%r")
print(CT_str12)

CT_str24=CT.strftime("%T")
print(CT_str24)

CDT_str =CDT.strftime("%d/%b/%Y %H:%M:%S")
print(CDT_str)
```

```
2022-10-08
10:39:08.895501
10:39:08 AM
10:39:08
08/Oct/2022 00:00:00
```

In [3]:

```
# 3. Python | Find yesterday’s, today’s and tomorrow’s date

import datetime
today = datetime.date.today()
yesterday = today - datetime.timedelta(1)
tomorrow = today + datetime.timedelta(1)
print('Yesterday : ',yesterday)
print('Today : ',today)
print('Tomorrow : ',tomorrow)

 # timedelta class from datetime module to find the previous day date and next day
```

```
Yesterday :  2022-10-07
Today :  2022-10-08
Tomorrow :  2022-10-09
```

In [6]:

```
# 8. How to convert timestamp string to datetime object in Python?

import datetime

date_time_str = '2018-06-29 08:15:27.243860'
x = datetime.datetime.strptime(date_time_str, '%Y-%m-%d %H:%M:%S.%f')

print('Date:', x.date())
print('Time:', x.time())
print('Date-time:', x)
```

```
Date: 2018-06-29
Time: 08:15:27.243860
Date-time: 2018-06-29 08:15:27.243860
```

In [2]:

```
# Tip and Trick 1: How to measure the time elapsed to execute your code in Python:-

import time
startTime = time.time()

time.sleep(7)
endTime = time.time()

totalTime = endTime - startTime

print("Total time required to execute code is= ", totalTime)
```

```
Total time required to execute code is=  7.003079175949097
```

In [11]:

```
# Getting formatted time:-
```

```
import time
  #returns the formatted time

print(time.asctime())
```

```
Sat Oct  8 11:54:10 2022
```

In [74]:
```
# Tip and Trick 2: Get the difference between the two Lists:-

list1 = ['Scott', 'Eric', 'Kelly', 'Emma', 'Smith']
list2 = ['Scott', 'Eric', 'Kelly']

x = set(list1)
y = set(list2)



z = x.intersection(y)         # &
s=x.symmetric_difference(y)   # ^
h=x.union(y)                  # |

print(z)
print(s)
print(h)
```

```
{'Kelly', 'Scott', 'Eric'}
{'Emma', 'Smith'}
{'Kelly', 'Emma', 'Smith', 'Eric', 'Scott'}
```

In [7]:
```
# Tip and Trick 5: Find if all elements in a list are identical:-

A = [20, 20, 20, 20]
y=set(A)
if len(y)==1:
    print("All element are duplicate in listOne")
else:
    print("No")

```

```
All element are duplicate in listOne
```

In [62]:
```
# Tip and Trick 6: How to efficiently compare two unordered lists:-

from collections import Counter

one = [33, 22, 11, 44, 55]
two = [22, 11, 44, 55, 33]

print("is two list are b equal", Counter(one) == Counter(two))

# or

print(sorted(one)==sorted(two))


#________________________________________

x = [33, 22, 11, 44, 55]
y = [22, 11, 44, 55, 33]

print(y==x)

for i in range(len(x)):
        for j in range(i+1,len(x)):
                if x[i]>x[j]:
                        x[i],x[j]=x[j],x[i]
print(x)

for i in range(len(y)):
        for j in range(i+1,len(y)):
                if y[i]>y[j]:
                        y[i],y[j]=y[j],y[i]
print(y)



print(y==x)
```

```
is two list are b equal True
True
False
[11, 22, 33, 44, 55]
[11, 22, 33, 44, 55]
True
```

In [42]:

```
# Tip and Trick 8: Use enumerate:-

listOne =[33, 22, 11, 44, 55]
print("Using enumerate")
for index, element in enumerate(listOne):
    print("Index [", index,"] Value", element)
```

```
Using enumerate
Index [ 0 ] Value 33
Index [ 1 ] Value 22
Index [ 2 ] Value 11
Index [ 3 ] Value 44
Index [ 4 ] Value 55
```

In [7]:

```
# Tip and Trick 9: Merge two dictionaries in a single expression:-

x = {1: 'Scott', 2: "Eric", 3:"Kelly"}
y  = {2: 'Eric', 4: "Emma"}

allEmployee = {**x, **y}
print(allEmployee)
```

```
{1: 'Scott', 2: 'Eric', 3: 'Kelly', 4: 'Emma'}
```

In [64]:

```
# Tip and Trick 10: Convert two lists into a dictionary:-

x= [54, 65, 76,6,7]
y = ["Hard Disk", "Laptop", "RAM","d"]

itemDictionary = dict(zip(y,x))

print(itemDictionary)
```

```
{'Hard Disk': 54, 'Laptop': 65, 'RAM': 76, 'd': 6}
```

In [10]:

```
# Tip and Trick 11: Convert hex string, String to int:-

hexNumber = "0xfdb"
stringNumber="34"

print("Hext to int", int(hexNumber,0))
print("String to int", int(stringNumber))
```

```
Hext to int 4059
String to int 34
```

In [84]:
```
# Tip and Trick 13: Return multiple values from a function:-

def multiplication_Division(num1,num2):
    return num1*num2, num1/num2, num2//num1

x,y,z=multiplication_Division(10,20)
print("mul:-",x, "Div:-", y,"Divisi",z)
```

```
mul:- 200 Div:- 0.5 Divisi 2
```

In [11]:
```
# # Tip and Trick 14:  order a list of numbers without built-in sort, min, max func

number = [64, 25, 12, 22, 11, 1,2,-44,3,122, 23, 34]

for i in range(len(number)):
    for j in range(i + 1, len(number)):

        if number[i] < number[j]:
           number[i], number[j] = number[j], number[i]
print (number)
```

```
[122, 64, 34, 25, 23, 22, 12, 11, 3, 2, 1, -44]
```

In [19]:
```
# Python sleep time
                # The sleep() method of time module is used to stop the execution o
                # The output will be delayed for the number of  seconds provided as

import time
for i in range(0,5):
    print(i)
    #Each element will be printed after 1 second
    time.sleep(3)

```

```
0
1
2
3
4
```

In [20]:
```
# The datetime Module:-

import datetime
#returns the current datetime object
print(datetime.datetime.now())
```

```
2022-06-11 11:17:33.001340
```

In [1]:
```
# Creating date objects:- We can create the date objects bypassing the desired date
                         # the date objects are to be created.

import datetime
#returns the datetime object for the specified date
print(datetime.datetime(2020,4,4))



#returns the datetime object for the specified time

print(datetime.datetime(2020,4,4,1,26,40))
```

```
2020-04-04 00:00:00
2020-04-04 01:26:40
```

In [16]:

```
# The calendar module :- Python provides a calendar object that contains various me
  # Consider the following example to print the calendar for the last month of 201

import calendar;
cal = calendar.month(2018,3)
#printing the calendar of December 2018
print(cal)
```

```
     March 2018
Mo Tu We Th Fr Sa Su
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

In [18]:

```
# Printing the calendar of whole year:- prcal() method of calendar module is used t

import calendar
#printing the calendar of the year 2020
s = calendar.prcal(2020)
```

```
                                  2020

      January                   February                   March
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                      1  2                         1
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       2  3  4  5  6  7  8
13 14 15 16 17 18 19      10 11 12 13 14 15 16       9 10 11 12 13 14 15
20 21 22 23 24 25 26      17 18 19 20 21 22 23      16 17 18 19 20 21 22
27 28 29 30 31            24 25 26 27 28 29         23 24 25 26 27 28 29
                                                    30 31

       April                      May                       June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                   1  2  3       1  2  3  4  5  6  7
 6  7  8  9 10 11 12       4  5  6  7  8  9 10       8  9 10 11 12 13 14
13 14 15 16 17 18 19      11 12 13 14 15 16 17      15 16 17 18 19 20 21
20 21 22 23 24 25 26      18 19 20 21 22 23 24      22 23 24 25 26 27 28
27 28 29 30               25 26 27 28 29 30 31      29 30

        July                     August                  September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
       1  2  3  4  5                      1  2          1  2  3  4  5  6
 6  7  8  9 10 11 12       3  4  5  6  7  8  9       7  8  9 10 11 12 13
13 14 15 16 17 18 19      10 11 12 13 14 15 16      14 15 16 17 18 19 20
20 21 22 23 24 25 26      17 18 19 20 21 22 23      21 22 23 24 25 26 27
27 28 29 30 31            24 25 26 27 28 29 30      28 29 30
                          31

      October                   November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su
          1  2  3  4                         1          1  2  3  4  5  6
 5  6  7  8  9 10 11       2  3  4  5  6  7  8       7  8  9 10 11 12 13
12 13 14 15 16 17 18       9 10 11 12 13 14 15      14 15 16 17 18 19 20
19 20 21 22 23 24 25      16 17 18 19 20 21 22      21 22 23 24 25 26 27
26 27 28 29 30 31         23 24 25 26 27 28 29      28 29 30 31
                          30
```

In [6]:

```
# Exercise 4: Reverse Dictionary mapping
```

```
x = {'A': 65, 'B': 66, 'C': 67, 'D': 68}
# Reverse mapping
new_dict = {value: key for key, value in x.items()}
print(new_dict)

#________________________________________

# Sort dict / By using dict comprehension:-

# sort by keys:-

dict1 = {"F":1,"C":3,"E":5,"D":2,"B":4,"A":6}

d = {key:dict1[key] for key in sorted( dict1 )}
print(d)


# Sort by Values
dict1 = {"F":1,"C":3,"E":5,"D":2,"B":4,"A":6}
d={value:key for key,value in dict1.items()}
print(d)
d1 = {key:d[key] for key in sorted( d )}
print(d1)
d2={value:key for key,value in d1.items()}
print(d2)
```

```
{65: 'A', 66: 'B', 67: 'C', 68: 'D'}
```

In [15]:

```
# Exercise 5: Display all duplicate items from a list:-

# # Solution 1: -

numbers = [10, 20, 60, 30, 20, 40, 30, 60, 70, 80]

duplicates = [i for i in numbers if numbers.count(i) > 1]
print(duplicates)


unique_duplicates = list(set(duplicates))

print(unique_duplicates)

#____________________________________________________________________

from collections import Counter


l1 = [1,2,2,3,4,5,2,5,6,7,8,9,9]
d = Counter(l1)
print(d)

new_list = ([i for i in d if d[i]>1])
print(new_list)
```

```
[20, 60, 30, 20, 30, 60]
[20, 60, 30]
Counter({2: 3, 5: 2, 9: 2, 1: 1, 3: 1, 4: 1, 6: 1, 7: 1, 8: 1})
[2, 5, 9]
```

In [10]:

```
# Exercise 6: Filter dictionary to contain keys present in the given list

# Dictionary
d1 = {'A': 65, 'B': 66, 'C': 67, 'D': 68, 'E': 69, 'F': 70}
# Filter dict using following keys
```

```
l1 = ['A', 'C', 'F']
new_dict = {key: d1[key] for key in l1}
print(new_dict)
```

```
{'A': 65, 'C': 67, 'F': 70}
```

In [14]:

```
# Exercise 9: Modify the element of a nested list inside the following list
# Change the element 35 to 400

list1 = [5, [10, 15, [20, 25, [30, 35], 40], 45], 50]
# modify item
list1[1][2][2][1] = 400
# print final result
print(list1[1])

# print(list1[1]) = [10, 15, [20, 25, [30, 400], 40], 45]
# print(list1[1][2]) = [20, 25, [30, 400], 40]
# print(list1[1][2][2]) = [30, 40]
# print(list1[1][2][2][1]) = 40
```

```
[10, 15, [20, 25, [30, 400], 40], 45]
```

In [6]:

```
# Python Program for Linear Search(To find index number)

arr = [1,2,3,4,5,6,7,8]
x = 4

for i,j in enumerate(arr):
    if j==x:
        print(i)
```

```
3
```

In [19]:

```
# Key Differences Between Recursion and Iteration?

# Recursion and iteration are both different ways to execute a set of instructions
# The main difference between these two is that in Recursion? :- The process in wh
# calls itself directly or indirectly is called recursion and the corresponding fu

# while in iteration, we use loops like "for" and "while" to do the same.
# Iteration is faster and more space-efficient than recursion.


# Factorial of a number using recursion

def factorial(n):
    if (n==1 or n==0):
        return 1
    else:
        return (n * factorial(n - 1))

#Driver Code
num = 5;
print("number : ",num)
print("Factorial : ",factorial(num))

#____________________________________________________________________________________

# Factorial of a number using iteration(for loop)
num=5
factorial=1
if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
```

```
else:
    for i in range(1,num+1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)

#___________________________________________________________________________________

#Factorial of a number using iteration(while loop)
def factorial(n):
    num=1
    while n>=1:
        num=num*n
        n=n-1
    return num

print("Factorial : ",factorial(5))
```

```
number :  5
Factorial :  120
The factorial of 5 is 120
Factorial :  120
```

In [11]:

```
# Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping t
# are in the wrong order.

number = [64, 25, 12, 22, 11, 1,2,-44,3,122, 23, 34]

for i in range(len(number)):
    for j in range(i + 1, len(number)):

        if number[i] < number[j]:
           number[i], number[j] = number[j], number[i]
print (number)


# Printing the first & second last element
print("Second largest element is:",[number[-2],number[-1]])
```

```
[122, 64, 34, 25, 23, 22, 12, 11, 3, 2, 1, -44]
Second largest element is: [1, -44]
```

In [22]:

```
# 4. Write a python program for string that will print out char with char count.

from collections import Counter

x="sHFGWGFFf JSFGgeg"

y = Counter(x)
print(y)
```

```
Counter({'F': 4, 'G': 3, 'g': 2, 's': 1, 'H': 1, 'W': 1, 'f': 1, ' ': 1, 'J': 1,
'S': 1, 'e': 1})
```

In [25]:

```
# Exercise 15: Use a loop to display elements from a given list present at odd inde
L = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
for i in L[1:-1:2]:     # or [1::2]
    print(i,end=",")
```

```
20,40,60,80,
```

In [29]:

```
# Exercise 16: Calculate the cube of all numbers from 1 to a given number:-

n=int(input("Enter Number:- "))
if n<1:
    print("Invalid Number")
```

```
for i in range(1,n+1):
    print(i**3)
```

```
Enter Number:- 5
1
8
27
64
125
```

In [ ]:

```
# Q9. Write a program to remove duplicate characters from the given input string?:

x='ABDQJHFFEABD'
l=[]
for i in x:
    if i not in l:
        l.append(i)
output=''.join(l)
print(output)
```

In [21]:

```
# Exercise 14,1: Program to reverse order of words.

s="Durga Software Solutions"
x=s.split(" ")
for i in x:
    print(i[::-1],end=" ")
print()

# Q4. Write a program to print characters at odd position and even position for the

s=input("Enter Some String:")
print("Characters at Even Position:",s[0::2])
print("Characters at Odd Position:",s[1::2])
```

```
agruD erawtfoS snoituloS
Enter Some String:vjk
Characters at Even Position: vk
Characters at Odd Position: j
```

In [20]:

```
# Exercise 12: Display Fibonacci series up to 10 terms:

num1,num2=0,1
for i in range(1,11):
    print(num1,end=" ")
    res=num1+num2
    num1=num2
    num2=res




#------------------------
# generat fibonaci  series by using A generator

def fib():
    a,b=0,1
    while True:
        yield a
        a,b=b,a+b
x= fib()
print(next(x))
print(next(x))
print(next(x))
print(next(x))
print(next(x))
print(next(x))


for f in fib():
    if f>10:
        break
    print(f,end=" ")
```

```
0 1 1 2 3 5 8 13 21 34
```

In [31]:
```
# Exercise 11: Write a program to display all prime numbers within a range:-
start = 25
end = 50
for i in range(start,end-1):
    if i>1:
        for j in range(2,i):
            if (i%j)==0:
                break
        else:
            print(i)
```

```
29
31
37
41
43
47
```

In [ ]:
```
# Exercise 10: Use else block to display a message “Done” after successful executio
for i in range(5):
    print(i)
else:
    print("Done")
```

In [ ]:
```
# Exercise 9: Display numbers from -10 to -1 using for loop:-

for i in range(-10,0,1):
    print(i)
```

In [24]:
```
# Exercise 2: Print the sum of the current number and the previous number
print("Printing current and previous number and their sum in a range(10)")

previous_num=0
for i in range(1,11):
    current_num=previous_num+i
    print("current number",i,"previous number",previous_num,"sum:-",i+previous_num
    previous_num=i
```

```
Printing current and previous number and their sum in a range(10)
current number 1 previous number 0 sum:- 1
current number 2 previous number 1 sum:- 3
current number 3 previous number 2 sum:- 5
current number 4 previous number 3 sum:- 7
current number 5 previous number 4 sum:- 9
current number 6 previous number 5 sum:- 11
current number 7 previous number 6 sum:- 13
current number 8 previous number 7 sum:- 15
current number 9 previous number 8 sum:- 17
current number 10 previous number 9 sum:- 19
```

In [30]:
```
# Exercise 3: Print characters from a string that are present at an even index numb
word="Amol D"
size=len(word)
```

```
print("printing only even char")
for i in range(0,size):
    if i%2==0:
        print(word[i])

print("printing only odd char")
for i in range(0,size):
    if i%2!=0:
        print(word[i])
```

```
printing only even char
A
o

printing only odd char
m
l
D
```

In [ ]:
```
# Exercise 4: Remove first n characters from a string
n=int(input("Enter n characters:- "))
word="Amol-Daund"
print(word[n: ])
```

In [ ]:
```
# Exercise 5: Check if the first and last number of a list is the same
x=[10,20,30,40,10]    # x="Amol A"
if x[0]==x[-1]:
    print("first and last number of a list is same")
else:
    print("first and last number of a list is not same")
```

In [ ]:
```
# Exercise 6: Display numbers divisible by 5 from a list
x=[10,23,45,68,233,400]
for i in x:
    if i%5==0:
        print(i)
```

In [ ]:
```
# Exercise 9: Check Palindrome Number
x=input("Enter number")
if (x==x[::-1]):
    print(x, "number is palindrome")
else:
    print(x, "number is not palindrome")

    # or

x=input("Check Palindrome Number : ")

if x==x[::-1]:
    print("Original Number: {0} \nYes. Given Number is palindrome number.".format(
else:
    print("Original Number: {0} \nNo. Given Number is not palindrome number.".form


```

In [ ]:
```
# Exercise 10: Create a new list from a two list using the following condition
 # the new list should contain odd numbers from the first list and even numbers fr
list1 = [10, 20, 25, 30, 35]
list2 = [40, 45, 60, 75, 90]
list=[]
for k in list1:
    if k%2!=0:
        list.append(k)
for k in list2:
    if k%2==0:
        list.append(k)
print(list)
```

In [ ]:
```
# Exercise 13: Print multiplication table form 1 to 10

for i in range(1,11):
    for j in range(1,11):
        print(i*j,end=" ")
    print()
```

In [ ]:
```
# Exercise 7: Return the count of a given substring from a string
x = "Emma is good developer. Emma is a writer"
a=x.count("Emma")
print(a)

#---------------------------------

x = "Emma is good developer. Emma is a writer"
y=x.split(" ")
z = set(y)
print(y)

for i in z:
    c=0
    for j in y:
        if i==j:
            c=c+1
    print(i,"occoured:-",c,"times")
```

In [ ]:
```
# Exercise 11: Write a Program to extract each digit from an integer in the reverse
b = 78459
a = str(b)
for i in range(len(a)-1,-1,-1):
    print(a[i],end='')
print()

# or

print(a[::-1])
```

In [ ]:
```
# Exercise 3: Convert Decimal number to octal using print() output formatting
x=28
print('%o' % x)  # % o - octal number base%
```

In [ ]:
```
# Exercise 4: Display float number with 2 decimal places using print()

num = 458.541315
print('%.2f' % num)

#___________________________________________

number= 88.2345
print('{:.2f}'.format(number))
```

In [ ]:
```
# Exercise 5: Accept a list of 5 float numbers as an input from the user

number=[]
# 5 is the list size
# run loop 5 times
```

```
for i in range(0, 5):
    print("Enter number at location", i, ":")
    # accept float number from user
    item = float(input())
    # add it to the list
    number.append(item)

print("User List:", number)
```

In [ ]:

```
# if else, for loop, and range() Exercises with Solutions:-

# Exercise 1: Print First 10 natural numbers using while loop:-
i=1
while i<=10:
    print(i)
    i=i+1
```

In [26]:

```
# Exercise 3: Calculate the sum of all numbers from 1 to a given number
n=int(input("Enter Number"))
print(sum(range(1,n+1)))

#----------------------------
p=0
for i in range(n+1):
    c=p+i
    p=c
print(c)

#______________________________

from functools import reduce
def add(x,y):
    return x+y

z = [5, 10, 15, 20, 25, 30, 35, 40, 45]

print(reduce(add,z))
```

```
Enter Number12
78
78
225
```

In [ ]:

```
# Exercise 4: Write a program to print multiplication table of a given number:-
n=int(input("Enter Number"))
for i in range(1,11):
    mul=n*i
    print(mul)
```

In [ ]:

```
# Exercise 5: Display numbers from a list using loop:-

# The number must be divisible by five
# If the number is greater than 150, then skip it and move to the next number
# If the number is greater than 500, then stop the loop


L=[12, 75, 150, 180, 145, 525, 50]
for i in L:
    if i>500:
        break
    elif i>150:
        continue
    elif i%5==0:
        print(i)
```

In [69]:
```
# Exercise 6: Count the total number of digits in a number:-

n=int(input("Enter Number"))
x=str(n)
print(len(x))
```

```
Enter Number243666
6
```

In [20]:
```
# Exercise 8: Print list in reverse order using a loop:-
L= [10, 20, 30, 40, 50]
R=reversed(L)
for i in R:
    print(i)

#_____________________________________________

L= [60, 70, 80, 90, 100]
for i in range(len(L)-1,-1,-1):
    print(L[i],end="\n")

#______________________________________________________

L= [60, 70, 80, 90, 100]


y=len(L)
print(y)


print(L[-1:-y-1:-1])

#________________________________________________

#input list
lst=[10, 11, 12, 13, 14, 15]

l=[] # empty list

# # checking if elements present in the list or not

for i in lst:
  #reversing the list
  l.insert(0,i)
print(l)
```

```
50
40
30
20
10
100
90
80
70
60
5
[100, 90, 80, 70, 60]
```

In [ ]:
```
# Exercise 2: Display three string “Name”, “Is”, “James” as “Name**Is**James”
```

```
x="My","Name","is","Amol"
print(*x,sep="**")
```

In [ ]:
```
# Exercise 15: Write a function called exponent(base, exp) that returns an int valu

def exponent(base, exp):
    calculate = base**exp
    return calculate

base = int(input('Enter the base: '))
exp = int(input('Enter the exponent: '))

print('The answer is', exponent(base, exp))
```

In [ ]:
```
# Exercise 1: Calculate the multiplication and sum of two numbers
   # Given two integer numbers return their product only if the product is greater
def mul_or_sum(num1,num2):
    product=(num1*num2)
    if product<=1000:
        return product
    else:
        return num1+num2

# first condition
x = mul_or_sum(50, 30)
print("The result is", x)

# Second condition
x = mul_or_sum(10, 30)
print("The result is", x)
```

In [7]:
```
# Formatting the Strings:
# Case- 1: Basic Formatting for default, positional and keyword arguments

name='durga'
salary=10000
age=48
print("{} 's salary is {} and his age is {}".format(name,salary,age))

# Case-2: Formatting Numbers:-

# d--->Decimal IntEger
# f----->Fixed point number(float).The default precision is 6
# b-->Binary format
# o--->Octal Format
# x-->Hexa Decimal Format(Lower case)
# X-->Hexa Decimal Format(Upper case)

print("The intEger number is: {}".format(123))
print("The intEger number is: {:d}".format(123))
print("The intEger number is: {:5d}".format(123))
print("The intEger number is: {:05d}".format(123))

print("The float number is: {}".format(123.4567))
print("The float number is: {:f}".format(123.4567))
print("The float number is: {:8.3f}".format(123.4567))
print("The float number is: {:08.3f}".format(123.4567))
print("The float number is: {:08.3f}".format(123.45))
print("The float number is: {:08.3f}".format(786786123.45))


# Eg-3: Print Decimal value in binary, octal and hexadecimal form
```

```
print("Binary Form:{0:b}".format(153))
print("Octal Form:{0:o}".format(153))
print("Hexa decimal Form:{0:x}".format(154))
print("Hexa decimal Form:{0:X}".format(154))
```

```
The intEger number is: 123
```

In [ ]:
```
# Python program to interchange first and last elements in a list

# Swap function
def swapList(newList):

    newList[0], newList[-1] = newList[-1], newList[0]

    return newList

# Driver code
newList = [12, 35, 9, 56, 24]
print(swapList(newList))
```

In [ ]:
```
# # Python code to replace, with . and vice-versa:-

def Replace(str1):
    str1 = str1.replace(', ', 'third')
    str1 = str1.replace('.', ', ')
    str1 = str1.replace('third', '.')
    return str1

string = "14, 625, 498.002"
print(Replace(string))
```

In [ ]:
```
# Python | Remove empty tuples from a list

tuples = [(), ('ram','15','8'), (), ('laxman', 'sita'),
          ('krishna', 'akbar', '45'), ('',''),()]

for i in tuples:
    if len(i)==0:
        tuples.remove(i)
print(tuples)
```

In [ ]:
```
# Python | Convert a list of Tuples into Dictionary

x = [('Key 1', 1), ('Key 2', 2), ('Key 3', 3), ('Key 4', 4), ('Key 5', 5)]
print(dict(x))
```

In [25]:
```
# Reverse while loop:-A

i = 10
while i >= 0:
    print(i, end=' ')
    i = i - 1
```

```
10 9 8 7 6 5 4 3 2 1 0
```

In [ ]:
```
# Example 3: Print even and odd numbers between 1 to the entered number.
n=int(input("enter number"))
while n>1:
    if n%2==0:
        print(n,"is even number")
    else:
        print(n,"is odd number")
    n=n-1
```

In [ ]:

```
# Python Program to check Armstrong Number: Armstrong number is a number that is ec

n=int(input("enter a number which you want to check if it is Armstrong:- "))
x=list(map(int,str(n)))    #  Convert each digit of the number to a string variable
y=list(map(lambda x:x**3,x))  # Then cube each of the digits and store in y variabl

print(y)
print(sum(y))

if(sum(y)==n):
    print("The",n," is an armstrong number. ")
else:
    print("The",n,"isn't an armstrong number. ")
```

In [ ]:

```
# Python | Ways to check if element exists in list

lst=[ 1, 6, 3, 5, 3, 4 ]

#checking if element 7 is present in the given list or not

i=7

# if element present then return exist otherwise not exist

if i in lst:
    print("exist")
else:
    print("not exist")

#------------------------------------------------------------------------------------

# Initializing list

test_list = [ 1, 6, 3, 5, 3, 4 ]

# Checking if 4 exists in list using loop

for i in test_list:
    if(i == 4) :
        print ("Element Exists")

#------------------------------------------------------------------------------------

# Checking if 4 exists in list
# using in
if (4 in test_list):
    print ("Element Exists")
```

In [33]:

```
# Different ways to clear a list in Python

# using clear() method

# Creating list
GEEK = [6, 0, 4, 1]
print('GEEK before clear:', GEEK)

# Clearing list
GEEK.clear()
print('GEEK after clear:', GEEK)
```

```
#________________________________________________

GEEK *=0
print('List1 after clearing using *= 0', GEEK)

#______________________________________
```

```
GEEK before clear: [6, 0, 4, 1]
GEEK after clear: []
List1 after clearing using *= 0 []
```

In [34]:
```
# Cloning or Copying a list Python


# Using the method of Shallow Copy

  # importing copy module
import copy

# initializing list 1
li1 = [1, 2, [3,5], 4]

# using copy for shallow copy
li2 = copy.copy(li1)

print(li2)


#-----------------------------------------------------------------------------------
# Using the method of Deep Copy
import copy

# initializing list 1
li1 = [1, 2, [3,5], 4]

# using deepcopy for deepcopy
li3 = copy.deepcopy(li1)
print(li3)
```

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

In [ ]:
```
# Count occurrences of an element in a list


from collections import Counter

# declaring the list
l = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5]

# driver program
x=3
d = Counter(l)
print('{} has occurred {} times'.format(x, d[x]))
```

In [ ]:
```
# Python | Multiply all numbers in the list


# list using lambda function and reduce()

from functools import reduce
list1 = [1, 2, 3]
list2 = [3, 2, 4]
```

```
result1 = reduce((lambda x, y: x * y), list1)
result2 = reduce((lambda x, y: x * y), list2)
print(result1)
print(result2)

#----------------------------------------------------------------------

import math
list1 = [1, 2, 3]
list2 = [3, 2, 4]

result1 = math.prod(list1)
result2 = math.prod(list2)
print(result1)
print(result2)
```

In [70]:
```
# Python program to print even numbers in a list

list1 = [10, 21, 4, 45, 66, 93]

# iterating each number in list
for num in list1:

    # checking condition
    if num % 2 == 0:
        print(num, end=" ")

#----------------------------------------------------------------------------------


list1 = [10, 21, 4, 45, 66, 93]

# using list comprehension
even_nos = [num for num in list1 if num % 2 == 0]

print("Even numbers in the list: ", even_nos)

#----------------------------------------------------------------------------------

list1 = [10, 21, 4, 45, 66, 93, 11]

# we can also print even no's using lambda exp.
even_nos = list(filter(lambda x: (x % 2 == 0), list1))

print("Even numbers in the list: ", even_nos)
```

```
10 4 66 Even numbers in the list:  [10, 4, 66]
Even numbers in the list:  [10, 4, 66]
```

In [ ]:
```
# Python program to print all even numbers in a range:-

for num in range(4,15,2):
    print(num,end=" \n")
```

In [ ]:
```
# Python program to count Even and Odd numbers in a List

list1 = [10, 21, 4, 45, 66, 93, 1]

even_count, odd_count = 0, 0

# iterating each number in list
for num in list1:
    # checking condition
    if num % 2 == 0:
        even_count += 1

    else:
        odd_count += 1

print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ", odd_count)

#--------------------------------------------------

list1 = [10, 21, 4, 45, 66, 93, 11]

even_count, odd_count = 0, 0
num = 0

# using while loop
while(num < len(list1)):

    # checking condition
    if list1[num] % 2 == 0:
        even_count += 1
    else:
        odd_count += 1

    # increment num
    num += 1

print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ", odd_count)
```

In [ ]:

```
# Python program to print positive numbers in a list

list1 = [11, -21, 0, 45, 66, -93]

# iterating each number in list
for num in list1:

    # checking condition
    if num >= 0:
        print(num, end = "\n")
#----------------------------------------------------------------

list1 = [-10, 21, -4, -45, -66, 93]
num = 0

# using while loop
while(num < len(list1)):

    # checking condition
    if list1[num] >= 0:
        print(list1[num], end = "\n")

    # increment num
    num += 1

#---------------------------------------------------------------------------------

list1 = [-10, -21, -4, 45, -66, 93]

# using list comprehension
pos_nos = [num for num in list1 if num >= 0]
```

```
print("Positive numbers in the list: ", *pos_nos)
```

In [ ]:
```
# Python program to count positive and negative numbers in a list

list1 = [10, -21, 4, -45, 66, -293, 1]

pos_count, neg_count = 0, 0

# iterating each number in list
for num in list1:

    # checking condition
    if num >= 0:
        pos_count += 1

    else:
        neg_count += 1

print("pos numbers in the list: ", pos_count)
print("neg numbers in the list: ", neg_count)

#---------------------------------------------------


list1 = [10, -21, 4, -45, 66, -93, -11]

pos_count, neg_count = 0, 0
num = 0

# using while loop
while(num < len(list1)):

    # checking condition
    if list1[num] >= 0:
        pos_count += 1
    else:
        neg_count += 1

    # increment num
    num += 1

print("pos numbers in the list: ", pos_count)
print("neg numbers in the list: ", neg_count)

#------------------------------------------------------

list1 = [-10, 21, 4, 45, 66, 93, 11]

neg_count = len(list(filter(lambda x: (x< 0) , list1)))

# we can also do len(list1) - neg_count
pos_count = len(list(filter(lambda x: (x> 0) , list1)))

print("pos numbers in the list: ", pos_count)
print("neg numbers in the list: ", neg_count)

#---------------------------------------------------------------------

list1 = [10, 21, 4, 45, 66, 93, 11]

only_neg = [num for num in list1 if num<0]
neg_count = len(only_neg)
```

```
print("pos numbers in the list: ", len(list1) - neg_count)
print("neg numbers in the list: ", neg_count)
```

In [ ]:
```
# Remove multiple elements from a list in Python

list1 = [11, 5, 17, 18, 23, 50]

# removes elements from index 1 to 4
# i.e. 5, 17, 18, 23 will be deleted
del list1[1:5]

print(*list1)

#---------------------------------------------

list1 = [11, 5, 17, 18, 23, 50]

# items to be removed
l2=list1[2:4]

list1 = [ele for ele in list1 if ele not in l2]

# printing modified list
print("New list after removing unwanted numbers: ", list1)
```

In [ ]:
```
# Python | Sort the values of first list using second list

x = ["O","X","A","C","D","K"]
y = ['1','2','3','4','5','6']

z = set(zip(x,y))
print(z)

for k,v in sorted(z):
    print(k,"=",v)
```

In [ ]:
```
# Python Ways to remove i’th character from string


test_str = "GeeksForGeeks"

# Removing char at pos 3
# using slice + concatenation
new_str = test_str[:3] +  test_str[4:]

print ("The string after removal of i'th character : " + new_str)
```

In [ ]:
```
# Find length of a string in python (4 ways)

str = "geeks"
print(len(str))

#-------------------------------

str ="My name is Amol"
count=0
for i in str:
    count+=1
print(count)

#-------------------------------
```

```
def findlen(str):
    c=0
    for i in str:
        c+=1
    return c
str="RamLakhan"
print(findlen(str))

#--------------------------------

def findlen(str):
    c=0
    while str[c:]:
        c+=1
    return c
str="RamLakhan"
print(findlen(str))
```

In [ ]:
```
# Python program to print even length words in a string
n="This is a python language"

s=n.split(" ")

for i in s:
  #checking the length of words
  if len(i)%2==0:
    print(i)

#---------------------------------------------

def Peven(n):
    s=n.split(' ')
    for i in s:
        if len(i)%2==0:
            return i

n="This is a python language"
print(Peven(n))
```

In [ ]:
```
# Python | Program to accept the strings which contains all vowels

a=input("name:- ")
v=0
c=0
for i in a:
    if (i=='a' or i=='e' or i=='i' or i=='o' or i=='u'or i=='A' or i=='E' or i=='I
        v=v+1
    else:
        c=c+1
print('T v:-', v)
print('T v:-', c)
```

In [ ]:
```
# Python program to count number of vowels using sets in given string Remove all du

a=input("name:- ")
x=set(a)
print(x)
v=0
for i in x:
    if (i=='a' or i=='e' or i=='i' or i=='o' or i=='u'or i=='A' or i=='E' or i=='I
        v=v+1
print('T v:-', v)
```

In [ ]:
```
# Python program to split and join a string

s = 'Brain works python'
# print the string after split method
y=s.split(" ")
print(y)

# print the string after join method
print("-".join(y))
```

In [ ]:
```
# Python | Check if a given string is binary string or not

string = "01010101010"
if(string.count('0')+string.count('1')==len(string)):
    print("Yes")
else:
    print("No")

#--------------------------------------------------------------------------------



x="01100"
try:
    y=int(x,2)     # convert binary to decimal
    print(y)
    print("Yes,The given string is binary")
except ValueError:
    print("The given string is not binary")
```

In [ ]:
```
# Python | Permutation of a given string using inbuilt function

from itertools import permutations

def allPermutations(str):

     # Get all permutations of string 'ABC'
    x = permutations(str)


     # print all permutations
    for i in list(x):
        print(''.join(i))


allPermutations("AmO")
```

In [66]:
```
# 7. write a program for character count string a5b3c2 for output: "aaabbbccaa"
s="a5b3c2"

op= ""

for ch in s:
        if ch.isalpha():
                x=ch
        else:
                d=int(ch)
                op=op+x*d
print(op)
```

```
aaaaabbbcc
```

In [67]:
```
s="7a4b3c2t"

op= ""

for ch in s:
        if ch.isdigit():
                x=int(ch)
        else:
                d=ch
                op=op+x*d
print(op)
```

```
aaaaaaabbbbccctt
```

In [ ]:
```
# Python program to find the sum of all items in a dictionary

def returnsum(dist):
    return sum(dist.values())

dict2 = {'a': 100, 'b': 200, 'c': 300}
print("Sum :", returnSum(dict2))
```

In [ ]:
```
# Python | Remove all duplicates words from a given sentence

string = 'Python is great and Java is also great'

print(' '.join(dict.fromkeys(string.split())))
```

In [ ]:
```
# Remove the items that are duplicated in two lists
list_1 = [1, 2, 1, 4, 6]
list_2 = [7, 8, 2, 1]
print(list(set(list_1)))
print(list(set(list_1) ^ set(list_2)))
```

In [ ]:
```
# Reverse the string

x='amol dai'
str=''
for i in x:
    str=i+str
print(str)
```

In [ ]:
```
# sqr of list element

x=[1,2,3,4,5]
s=list(map(lambda i:i**i,x))
print(s)
```

In [ ]:
```
# add all list of element

from functools import reduce

def add(x,y):
    return x + y
list1=[1,2,3,4,5,6,7,8,9,10]
print(reduce(add,list1))
```

In [ ]:
```
# access the key and values

x={1:"a",2:"s"}
print(x.keys())
print(x.values())
print(type(x.keys()))
```

In [40]:
```
# Flattening a multi-dimensional list


ML = [[10,20,30],[40,50,60],[70,80,90]]
L = [x for i in ML for x in i]
print(L)
```

```
[10, 20, 30, 40, 50, 60, 70, 80, 90]
```

In [ ]:
```
# Combining multiple lists into one


a = [1, 2, 3]
b = [7, 8, 9]
y= [(x + y) for (x,y) in zip(a,b)]  # parallel iterators
print(y)

z=[(x,y) for x in a for y in b]    # nested iterators
print(z)
```

In [43]:
```
# Printing the elements of the list with its index number using the range() functio

numbers = [1, 2, 4, 6, 8]
size = len(numbers)
for i in range(size):
    print('Index:', i, " ", 'Value:', numbers[i])
```

```
Index: 0   Value: 1
Index: 1   Value: 2
Index: 2   Value: 4
Index: 3   Value: 6
Index: 4   Value: 8
```

In [ ]:
```
# Example 1: Check how many times a given number can be divided by 3 before it is l
x=0
number=180
while number > 10:
    number=number/3
    x=x+1
print("Total iteration required",x)
```

In [60]:
```
# How to enable and disable Garbage Collector in our program:
# By default Gargbage collector is enabled, but we can disable based on our require
# context we can use the following functions of gc module.
# 1. gc.

import gc
gc.isenabled()

# 2. gc.disable()
# To disable GC explicitly
# 3. gc.enable()
# To enable GC explicitly
```

In [ ]:
```
# 9. Problem: Remove specified characters in a string irrespective of the
# case.char_to_remove =['A','N'] string= 'Think Analytics'
```

In [ ]:
```
# prime number
for n in range(1,1000):
        s = 0
        for i in range(1, n):
                if n % i == 0:
                        s=s+i
        if s==n:
                print({s},"is a prime number")

        else:
                print({n},"is not a prime number")
```

In [ ]:
```
# Destructors:
# Destructor is a special method and the name should be __del__
# Just before destroying an object Garbage Collector always calls destructor to per
# activities (Resource deallocation activities like close database connection etc)
# Once destructor execution completed then Garbage Collector automatically destroys

# Note: The job of destructor is not to destroy object and it is just to perform cl

import time
class Test:
    def __init__(self):
        print("Object Initialization...")
    def __del__(self):
        print("Fulfilling Last Wish and performing clean up activities...")

t1=Test()
t1=None
time.sleep(5)

print("End of application")
```

In [ ]:
```
# split         Returns a list where the string has been split at each match#

import re

txt = "The rain in Spain"

x = re.split("\s", txt,2)    # 1 :Split the string only at the first occurrence:
print(x)
```

In [ ]:
```
import re, os

# list of different types of file
filenames = r'C:\Users\DELL\02.JN_Projects\JN_Projects'

for file in os.listdir(filenames):
    # search given pattern in the line
    match = re.search("\.txt$", file)

    # if match is found
    if match:
        print("The file ending with .xml is:",file)
```

In [ ]:
```
#  find the pair with given number in a list:-

L=[1,2,4,5,3,8,7,6]
n=len(L)

k=10
```

```
for i in range(n):
    for j in range(i,n):      # here (i,n) for accending order and (n) for decendi
        if L[i]+L[j]==k:
            print(L[i],L[j])
```

In [ ]:

```
# Write a code to raise an exception:-

L=[1,2,4,5,3,7,6]
# L=[2,4,5,3,7,6]
sum=0
for i in L:
    if i==1:
        raise Exception("Exception: 1 bis found")
    else:
        sum+=i
print(sum)
```

In [ ]:

```
x="Amol "
y="Daund"
x=list(x)
y=list(y)

z=list(map(lambda x,y:x+y,x,y))
print("".join(z))
```

In [76]:

```
# Example: Write a program to print the table of the given number using the genera

def table(n):
    for i in range(1,11):
        yield n*i
        i = i+1

for i in table(4):
    print(i)
```

```
4
8
12
16
20
24
28
32
36
40
```

In [ ]:

```
# create a Student class and Creates an object to it. Call the method Details() to

class Student:

    def __init__(self,name,rollno,marks):
        self.name=name
        self.rollno=rollno
        self.marks=marks

    def Details(self):
        print("Student Information:- \nName:{} \nRollno:{} \nMarks:{}".format(self

s1=Student("Amol",5,35)
s1.Details()

s2=Student("Sagar",5,35)
```

```
s2.Details()
print(s2.__dict__)
```

In [ ]:

In [ ]: