In [1]:
```
#Prg1
#addition of two numbers#method1
a =20
b =15
sum = a+b
print(sum)
```

```
35
```

In [4]:
```
#addition of two numbers#method2
a =int(input("enter the no1:"))#get nos from users
b =int(input("enter the no2:"))
sum = a+b
print(sum)
```

```
enter the no1:5
enter the no2:8
13
```

In [6]:
```
#prg2
#maximum two numbers# method1
a = 5
b=9
maximum =max(a,b)
print(maximum)
```

```
9
```

In [7]:
```
def maximum(a, b):#method2 for max

    if a >= b:
        return a
    else:
        return b
print(max(10,3))
```

```
10
```

In [8]:
```
#method3 for max
a=3
b=5
print(a if a >= b else b)
```

```
5
```

In [12]:
```
#prg3
#factorial prg#method1
def factorial(n):

      return 1 if (n==1 or n==0) else n * factorial(n - 1);
num = 5
print("Factorial of num is",num)
factorial((num))
```

```
Factorial of num is 5
```

Out[12]:
```
120
```

In [15]:
```
#method2
def factorial(n):
    if n < 0:
        return 0
    elif n == 0 or n == 1:
        return 1
    else:
        fact = 1
        while(n > 1):#n=5
            fact *= n
            n -= 1#decrement
        return fact

num =7
print("Factorial of no is",num)
factorial((num))
```

Out[15]:
```
Factorial of no is 7
5040
```

In [16]:
```
def factorial(n):#recusive function

    # single line to find factorial
    return 1 if (n==1 or n==0) else n * factorial(n - 1)

num = 5
print ("Factorial of no is",num)
print(factorial(num))
```

```
Factorial of no is 5
120
```

In [18]:
```
#method 4
import math

def factorial(n):
    return(math.factorial(n))#using inbuilt function

num = 5
print("Factorial of no is", num)
print(factorial(num))
```

```
Factorial of no is 5
120
```

In [19]:
```
#prg4
#find simple interest
def simple_interest(p,t,r):
    print('The principal is', p)
    print('The time period is', t)
    print('The rate of interest is',r)

    si = (p * t * r)/100#formula for si

    print('The Simple Interest is', si)
    return si

simple_interest(8, 6, 8)
```

```
The principal is 8
The time period is 6
The rate of interest is 8
The Simple Interest is 3.84
```

Out[19]:
```
3.84
```

In [20]:
```
#prg5
#area of the circle(pir2)
def findArea(r):
    PI = 3.142#constant
    return PI * (r*r)

print("Area is %.6f" % findArea(5))#0.6f denotes 6 digits after decimal
```

```
Area is 78.550000
```

In [21]:
```
#prg6
#compound interest
def compound_interest(principle, rate, time):

    # Calculates compound interest
    Amount = principle * (pow((1 + rate / 100), time))#A = P*(1 + R/100)powe
    CI = Amount - principle
    print("Compound interest is", CI)

compound_interest(10000, 10.25, 5)
```

```
Compound interest is 6288.946267774416
```

In [22]:
```
#prg7
#ascii values#method1
c = 'g'
# print the ASCII value of assigned character in c
print("The ASCII value of '" + c + "' is", ord(c))
```

```
The ASCII value of 'g' is 103
```

In [2]:
```
#method2
text = input("enter the string")
for char in text:
    ascii = ord(char)
    print(char, "\t", ascii)
```

```
enter the stringpython
p        112
y        121
t        116
h        104
o        111
n        110
```

In [29]:
```
#prg8 #SQUARe and sum
def squaresum(n) :

    sm = 0
    for i in range(1, n+1) :
        sm = sm + (i * i)
    return sm
print(squaresum(4))
```

```
30
```

In [30]:
```
#method2
def squaresum(n):
    return (n * (n + 1) / 2) * (2 * n + 1) / 3#using formula

n = 4
print(squaresum(n));
```

```
30.0
```

In [31]:
```
#prg9
#sum of cube numbers#method 1 direct

def cubessum(n):
    sum = 0
    for i in range(1, n+1):
        sum +=i*i*i

    return sum
print(cubessum(5))
```

```
225
```

In [33]:
```
def sumOfSeries(n):#method2 using formula
    x = (n * (n + 1)  / 2)
    return (int)(x * x)#formula((n*n+1)/2 )power of 2
n = 8
print(sumOfSeries(n))
```

```
1296
```

In [8]:
```
#prg10#find pos
def findPosition(k, n):
    f1 = 0
    f2 = 1
    i =2;
    while i!=0:
        f3 = f1 + f2;
        f1 = f2;
        f2 = f3;

        if f2%k == 0:
            return n*i

        i+=1

    return


# Multiple no.
n = 5
# Number of whose multiple we are finding
k = 4

print(findPosition(k,n))
```

```
30
```

In [18]:
```
#prg11
#to find sum of array#method1
def _sum(arr):

    sum=0 #to store the sum

    for i in arr:#iterate array
        sum = sum + i

    return(sum)

# input values to list
arr = [12, 3, 4, 15]
#function call

# display sum
print ('Sum of the array is ',_sum(arr))
```

```
Sum of the array is  34
```

In [19]:
```
arr = [12, 3, 4, 15]#method2 using inbuilt fun
ans = sum(arr)

# display sum
print ('Sum of the array is ',ans)
```

```
Sum of the array is  34
```

In [28]:
```
#prg12
#FIND LARGEST NUMBER IN ARRAY
def largest(arr,n):

    max = arr[0]#assigned o index is max
    for i in range (1,n):#iterating fdrom index1 to n
        if arr[i]> max:
            max = arr[i]
    return max
arr = [10,50,70,40,30]
n =len(arr)
print("largest number is:",largest(arr,n))
```

```
largest number is: 70
```

In [38]:
```
#prg 13
#remainder of array#not clear having doubt
array= [100,10,5,25,35,14]
n=len(array)
rem =0
mul =1
for i in range(1,n):
    rem = array[i]%n
    mul=mul*rem
rem1=mul%n
print(rem1)
```

```
2
```

In [44]:
```
#prg14
#lswap two elements#method1

def swap_pos(list,p1,p2):
    list[p1],list[p2]=list[p2],list[p1]
    return list

list=[10,20,50,70]
p1,p2=0,2
print(swap_pos(list,p1,p2))
```

```
[50, 20, 10, 70]
```

In [48]:
```
#method2
def swp_Pos(list,p1,p2):
    get = list[p1],list[p2]
    list[p2],list[p1] = get
    return list
list = [10,20,30,40]
p1,p2=1,3
print(swap_pos(list,p1,p2))
```

```
[10, 40, 30, 20]
```

In [51]:
```
#prg15
#length of list#method1
list1= [19,57,89,34,12,11,45]
print("the input list is",list1)
#using loop
c=0#for storing count
for i in list1:
    c=c+1
print("the length of the list is",c)
```

```
the input list is [19, 57, 89, 34, 12, 11, 45]
the length of the list is 7
```

In [64]:
```
#method2 using len()
a=[]
a.append("hi")
a.append("welcome")
a.append("to")
a.append("python")
res = len(a)
print("the length of the list is:",res)
```

```
the length of the list is: 4
```

In [65]:
```
#method3
n = len([10, 20, 30])
print("The length of list is: ", n)
```

```
The length of list is:  3
```

In [75]:
```
#prg16
#check ements inlist
#method1
list=[1,6,3,5,3,4]
if(4 in list):
    print("the 4 is present")
#method2
for i in list:
    if i==4:
        print("elments exists")
#method3 usingset and in
list1=set(list)
if (4 in list1):
    print("4 is present in set")
#method4 using count method
count1 = list.count(6)
if count1>0:
    print("the 6 is present")
```

```
the 4 is present
elments exists
4 is present in set
the 6 is present
```

In [85]:
```
#prg17
#list clear method#using builtin fun
list=[20,30,40,11,34,56]
list.clear()
print(list)

#method2
l1 = [1,2,3]
l2=[3,5,7,9]
l1.clear()
l2=[]#l2 clear using assigning method
print(l1,l2)
#method3
l =[3,5,6,1,6]
list *= 0
print(l)
#method4 using del
list1=[3,9,4,6,3]
del list1[:]
print(list1)
```

```
[]
[] []
[3, 5, 6, 1, 6]
[]
```

In [1]:
```
#prg18 reverese a list
def Reverse(lst):
    return [ele for ele in reversed(lst)]
lst = [10, 11, 12, 13, 14, 15]
print(Reverse(lst))
#method2
lst = [10, 11, 12, 13, 14, 15]
rev = lst.reverse()#usong built in fun
print(lst)
#method3 using slicing concept
lst1=[10, 11, 12, 13, 14, 15]
nw_lst1=lst1[::-1]
print(nw_lst1)
```

```
[15, 14, 13, 12, 11, 10]
[15, 14, 13, 12, 11, 10]
[15, 14, 13, 12, 11, 10]
```

In [ ]:

In [2]:
```
#prg19
#Python program to find sum of elements in list
sum1=0
list1 = [11, 5, 17, 18, 23]
n = len(list1)
for ele in range(0,n) :
    sum1 = sum1+list1[ele]
print("sum of the list is",sum1)

#method2 using while loop
list1 = [11, 5, 17, 18, 23]
total=0
i=0
while i<len(list1):
    total = total+list1[i]
    i+=1
print("sum of the list is",total)

#method3 using built in fun
list1 = [11, 5, 17, 18, 23]
print("sum of the list is",sum(list1))
```

```
sum of the list is 74
sum of the list is 74
sum of the list is 74
```

In [ ]:
```
#prg20 multiply the list
def multiply(mylist):
    mul=1
    for i in mylist:
        mul = mul*i
    return mul

list1 = [1, 2, 3]
list2 = [3, 2, 4]
print(multiply(list1))
print(multiply(list2))
#method2
# 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)#reduce used like argument in f
result2 = reduce((lambda x, y: x * y), list2)
print(result1)
print(result2)

#method3
import math#using module
list1 = [1, 2, 3]
list2 = [3, 2, 4]

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

In [ ]:
```
#prg21#find smallest no using sort method
list1 = [10, 20, 4, 45, 99]
list1.sort()
small = list1[:1]#slicing method
print("the small number is:",small)

#method2#using built in fun
list1 = [10, 20, 1, 45, 99]
small = min(list1)
print("the small number is:",small)

#method 3
lst=[]
n=int(input("enter the no of element"))
for i in range(1,n+1):
    int1 = int(input())
    lst.append(int1)
print("the small number is:",min(lst))
```

In [2]:
```
#prg22#find largest no using sort method
list1 = [10, 20, 4, 45, 99]
list1.sort()
large = list1[-1:]#slicing method
print("the largest number is:",large)

#method2#using built in fun
list1 = [10, 20, 1, 45, 99]
large = max(list1)
print("the largest number is:",large)

#method 3#list appen method
lst=[]
n=int(input("enter the no of element"))
for i in range(1,n+1):
    int1 = int(input())
    lst.append(i)
print("the largest number is:",max(lst))
```

```
the largest number is: [99]
the largest number is: 99
enter the no of element4
1
3
7
9
the largest number is: 4
```

In [2]:
```
#prg 23
#method1
# Python program to print Even Numbers in a List

# list of numbers
list1 = [10, 21, 4, 45, 66, 93]
for i in list1:
    if i%2==0:
        print(i,end=' ')
```

```
10 4 66
```

In [1]:
```
#method2
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)
```

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

In [ ]:
```
#method3
list1 = [10, 24, 4, 45, 66, 93]
num = 0

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

     if list1[num] % 2 == 0:
        print(list1[num], end=" ")
        num += 1
```

```
10 24 4
```

In [ ]:
```
# method4
list1 = [10, 21, 4, 45, 66, 93, 11]
even_nos = list(filter(lambda x: (x % 2 == 0), list1))#using lamda

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

In [ ]:

In [ ]: