1. Manipulate using a list

```
In [1]: # to add new elements to end of the list

        List=['2','5','7','5','2','6','3']
        List

Out[1]: ['2', '5', '7', '5', '2', '6', '3']
```

```
In [2]: List.append('9') # 9 is added in the list by using append
```

```
In [3]: List

Out[3]: ['2', '5', '7', '5', '2', '6', '3', '9']
```

```
In [4]: List.extend('10')  # 10 is added in the list by using extend
```

```
In [5]: List

Out[5]: ['2', '5', '7', '5', '2', '6', '3', '9', '1', '0']
```

```
In [6]: # to reverse elements in the list:
        List.reverse()
        print(List.reverse)

        <built-in method reverse of list object at 0x000001971CCAFAC0>
```

```
In [7]: List

Out[7]: ['0', '1', '9', '3', '6', '2', '5', '7', '5', '2']
```

```
In [8]: # to reverse elements in the list:
        List.reverse()
        print(List)

        ['2', '5', '7', '5', '2', '6', '3', '9', '1', '0']
```

```
In [9]: List

Out[9]: ['2', '5', '7', '5', '2', '6', '3', '9', '1', '0']
```

```
In [10]: List.reverse() # used to reverse the elements
```

```
In [11]: List

Out[11]: ['0', '1', '9', '3', '6', '2', '5', '7', '5', '2']
```

```
In [12]: # to display the same list of elements multiple times :
         List

Out[12]: ['0', '1', '9', '3', '6', '2', '5', '7', '5', '2']
```

```
In [14]: repeated_List= List *2 # used to repeat the elements 2 times
         print(repeated.List)

         ---------------------------------------------------------------------------
         NameError                                 Traceback (most recent call last)
         Cell In[14], line 1
         ----> 1 repeated.List= List *2 # used to repeat the elements 2 times
               2 print(repeated.List)

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

```
In [15]: repeated_List= List *2 # used to repeat the elements 2 times
         print(repeated_List)

         ['0', '1', '9', '3', '6', '2', '5', '7', '5', '2', '0', '1', '9', '3', '6', '2', '5', '7', '5', '2']
```

In [17]:

```
# to concatenate two list:

List_1=['34','67','45','98']
List_2=['39','24','09','878']
```

In [18]:

```
concate_list= List_1 + List_2
```

In [19]:

```
print(concate_list)
```

```
['34', '67', '45', '98', '39', '24', '09', '878']
```

In [20]:

```
# to sort the elements in the list in ascending order:
List=[1,3,5,779,245,977,14625,98,2,78,9]
```

In [21]:

```
List.sort()
List
```

Out[21]:

```
[1, 2, 3, 5, 9, 78, 98, 245, 779, 977, 14625]
```

In [22]:

```
List
```

Out[22]:

```
[1, 2, 3, 5, 9, 78, 98, 245, 779, 977, 14625]
```

# 2. Write a Python program to do in the tuples.

In [27]:

```
# manipulate using tuples
tuple1=(2,4,6,8,10)
tuple1.count()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 3
      1 # manipulate using tuples
      2 tuple1=(2.4,6,8,10)
----> 3 tuple1.count()

TypeError: tuple.count() takes exactly one argument (0 given)
```

In [31]:

```
# add new elements end of the tuple
tuple1=(2,4,6,8,10)
tup= tuple1 + (1,3,5,7,9)
tup
```

Out[31]:

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

tuple2

In [30]:

```
tuple2
```

Out[30]:

```
(1, 3, 5, 7)
```

In [32]:

```
tup
```

Out[32]:

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

In [33]:

```
# to reverse elements in the list
tup[::-1]
```

Out[33]:

```
(9, 7, 5, 3, 1, 10, 8, 6, 4, 2)
```

In [35]:

```
#to display the elements of the same tuple multiple times
tup*2
```

Out[35]:

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9, 2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

In [36]:

```
# to concatenate two tuples
tuple1=(2,4,6,8,10)
tuple2=(1,3,5,7,9)
concated_tuple= tuple1 + tuple2
concated_tuple
```

Out[36]:

```
(2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

```
In [37]: tup
```

```
Out[37]: (2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

```
In [38]: #to sort the elements inthe ascending order

         tup.sort()
         tup
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[38], line 3
      1 #to sort the elements inthe ascending order
----> 3 tup.sort()
      4 tup

AttributeError: 'tuple' object has no attribute 'sort'
```

```
In [42]: tuple(sorted(tup))
```

```
Out[42]: (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
```

```
In [44]: list(tup).sort()
```

```
In [40]: tuple(sort(tup))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[40], line 1
----> 1 tuple(sort(tup))

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

```
In [45]: tup
```

```
Out[45]: (2, 4, 6, 8, 10, 1, 3, 5, 7, 9)
```

# 3. Write a python program to implement the follwing using list

```
In [ ]: # create a list with integers (Minimum 10 numbers)
```

```
In [46]: list=[23,56,77,98,45,17,89,65,43,29]
         list
```

```
Out[46]: [23, 56, 77, 98, 45, 17, 89, 65, 43, 29]
```

```
In [47]: #how to display the last number in the list
         list[:-1]
```

```
Out[47]: [23, 56, 77, 98, 45, 17, 89, 65, 43]
```

```
In [48]: list[-1]
```

```
Out[48]: 29
```

```
In [49]: #displaying values from the list[0:4]
         list[0:4]
```

```
Out[49]: [23, 56, 77, 98]
```

```
In [50]: #displaying values from the list[2:]
         list[2:]
```

```
Out[50]: [77, 98, 45, 17, 89, 65, 43, 29]
```

```
In [51]: #displaying values from the list[:6]
         list[:6]
```

```
Out[51]: [23, 56, 77, 98, 45, 17]
```

```
In [52]: list[0:6]
```

```
Out[52]: [23, 56, 77, 98, 45, 17]
```

# 4. Write a python program tuple1=(10,50,20,40,30)

In [53]:
```
#to dsiplay the elements 10 and 50 from tuple1
t1=(10,50,20,40,30)
t1[:2]
```

Out[53]:
```
(10, 50)
```

In [54]:
```
# to display the length of the tuple1
t1.length()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[54], line 2
      1 # to display the length of the tuple1
----> 2 t1.length()

AttributeError: 'tuple' object has no attribute 'length'
```

In [55]:
```
len(t1)
```

Out[55]:
```
5
```

In [56]:
```
#to find the minimum element from tuple
min(t1)
```

Out[56]:
```
10
```

In [57]:
```
#to add all the elements
sum(t1)
```

Out[57]:
```
150
```

In [58]:
```
#to display the same tuple1 multiple time

t1*2
```

Out[58]:
```
(10, 50, 20, 40, 30, 10, 50, 20, 40, 30)
```

# 5. Write a python program.

In [59]:
```
#i length of the string
S='Balijepalli Sravan Kumar'
S
```

Out[59]:
```
'Balijepalli Sravan Kumar'
```

In [60]:
```
len(S)
```

Out[60]:
```
24
```

j

In [61]:
```
S.reverse()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[61], line 1
----> 1 S.reverse()

AttributeError: 'str' object has no attribute 'reverse'
```

In [62]:
```
reverse(S)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[62], line 1
----> 1 reverse(S)

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

```
In [63]: #2
         S[::-1]

Out[63]: 'ramuK navarS illapejilaB'

In [69]: #3
         S *4

Out[69]: 'Balijepalli Sravan KumarBalijepalli Sravan KumarBalijepalli Sravan KumarBalijepalli Sravan Kumar'

In [71]: #4
         S1='Hello'
         result = S1+" " + S
         result

Out[71]: 'Hello Balijepalli Sravan Kumar'

In [74]: #5
         Str1= "South India"
         Str1[6:11]

Out[74]: 'India'
```

# 6. Perform the following

```
In [76]: #1 Creating dictonary
         s_dict={'Name': 'Sravan Kumar', 'Employee Id': '2575263', 'Domain': 'AIML'}

In [77]: s_dict

Out[77]: {'Name': 'Sravan Kumar', 'Employee Id': '2575263', 'Domain': 'AIML'}

In [78]: #2 accesing values
         s_dict.items()

Out[78]: dict_items([('Name', 'Sravan Kumar'), ('Employee Id', '2575263'), ('Domain', 'AIML')])

In [79]: #3 Updating the dictionary using function
         s_dict.update{'age': '23'}

           Cell In[79], line 2
             s_dict.update{'age': '23'}
                          ^
         SyntaxError: invalid syntax



In [80]: s_dict.update({'age': '23'})

In [81]: s_dict

Out[81]: {'Name': 'Sravan Kumar',
          'Employee Id': '2575263',
          'Domain': 'AIML',
          'age': '23'}

In [82]: # clear and delete
         s_dict.clear()

In [83]: s_dict

Out[83]: {}
```

# 7. Python Program to insert a number to any position in a list

```
In [84]: list=[2,4,6,8]
         list

Out[84]: [2, 4, 6, 8]

In [85]: list.insert(0,1)
In [86]:  list
```

```
Out[86]: [1, 2, 4, 6, 8]
```

```
In [88]:  list.insert(2,3)
          list
```

```
Out[88]: [1, 2, 3, 4, 3, 6, 8]
```

```
In [89]:  list
```

```
Out[89]: [1, 2, 3, 4, 3, 6, 8]
```

# 8. Python Program to delete an element from a list by index

```
In [90]:  list=[1,2,3,4,7,5]
          list
```

```
Out[90]: [1, 2, 3, 4, 7, 5]
```

```
In [111]: list.pop(4)
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[111], line 1
----> 1 list.pop(4)

IndexError: pop index out of range
```

```
In [112]: list
```

```
Out[112]: ['hello', 'Dear', 'hOw', 'ARe']
```

```
In [113]: list1= [1,2,3,4,5,8,6]
          list1
```

```
Out[113]: [1, 2, 3, 4, 5, 8, 6]
```

```
In [98]:  list1.remove(6)
```

```
In [99]:  list1
```

```
Out[99]: [1, 2, 3, 4, 5, 8]
```

# 9. Write a program to display a number from 1 to 100

In [106]:
```
for i in range(1, 101):
    print(i)
```

```
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

# 10. Write a python program to find sum of all items in a tuple

In [107]:

```
list=(1,2,3,4,5,6,7,8,9,10)
sum(list)
```

Out[107]:

```
55
```

# 11. Create a dictionary containg three lambda functions square,cube,sqrtroot

In [ ]:

In [ ]:

# 12. A list of words is given. Find the words the list that have their second character in uppercase

In [109]:

```
list=['hello', 'Dear', 'hOw', 'ARe', 'You']
for i in list:
    if i[1].isupper():
        print(i)
```

```
hOw
ARe
```

# 13. A dictionary pf names and their weights on earth is given. Find how much they will weigh on the moon.

In [114]:

```
weight_on_earth={'john':45, 'Shelly': 65, 'Marry': 35}
weight_on_earth
```

Out[114]:

```
{'john': 45, 'Shelly': 65, 'Marry': 35}
```

In [115]:

```
weight_on_moon={}
for i,j in weight_on_earth.items():
    weight_on_moon[i] = ((j*1.622)/9.81)
```

In [116]:

```
weight_on_moon
```

Out[116]:

```
{'john': 7.440366972477065,
 'Shelly': 10.747196738022426,
 'Marry': 5.786952089704383}
```

# Control Structures

In [127]:

```
def Prime(n):
    for i in range(2,n//2+1):
        if(n%i==0):
            return(0)
    return(1)

N=int(input("Enter N:"))
i=2
lst=[]
while(1):
    if(Prime(i)):
        lst.append(i)
        if(len(lst)==N):
            break
    i+=1
print( "Prime numbers are: ",end="")
print(*lst)
```

```
Enter N:5
Prime numbers are:2 3 5 7 11
```

In [129]:

```
#2 Write the python code that calculates the salary the salary of an employee.

Basic_salary = int(input("Enter the basic salary"))
HRA=  int(input("Enter the basic HRA"))
TA= int(input("Enter the basic TA"))
DA= int(input("Enter the basic DA"))

Gross_salary= Basic_salary+HRA+TA+DA
tax= (10/100)*Gross_salary
Net_salary = Gross_salary - tax
print("Gross Salary: " , Gross_salary)
print("tax: " , tax)
print("Net Salary: " , Net_salary)
```

```
Enter the basic salary25000
Enter the basic HRA8000
Enter the basic TA2200
Enter the basic DA1700
Gross Salary:  36900
tax:  3690.0
Net Salary:  33210.0
```

In [130]:

```
#3write a python code to search for a string in the given list
Str1 = ['S','R','A','V','A','N','K','U','M','A','R']
s = input('Please enter a letter: ')

if s in Str1:
    print(f'{s} is present in the list')
else:
    print(f'{s} is not present in the list')
```

```
Please enter a letter: S
S is present in the list
```

In [139]:

```
#4Write a python function thataccepts s tring and caculates that number of upper letters and lower letters

S=('S','r','A','V','a','n')
upp=0
low=0

for i in S:
    if i.isupper():
        upp+=1
    else:
        low+=1

print("Number of upper letter are: ", upp)
print("Number of lower letter are: ", low)
```

```
Number of upper letter are:  3
Number of lower letter are:  3
```

In [134]:

```
#5 Write a program to display the sum of odd numbers and even numbers that fall between 12 and 37

odd = []
even = []
total = 0
for i in range(12,37):
    if(i%2 == 0):
        even.append(i)
    else:
        odd.append(i)
    total += i
print("Even numbers: ",even)
print("Odd numbers: ",odd)
print("Total is:{} ".format(total))
```

```
Even numbers:  [12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36]
Odd numbers:  [13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35]
Total is:600
```

In [141]:

```
#6 Write a python program to print the table of any number
s = int(input ("Enter the number: "))
print ("The Multiplication Table of: ", s)

for i in range(1, 11):
   print (s, 'x', i, '=', s * i)
```

```
Enter the number: 5
The Multiplication Table of:  5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

In [165]:

```
#7 Write a python program to sum the first 10 prime numbers

from math import sqrt

num = int(input("Enter a number: "))
count = 0
n = 2
l=[]

while count < num:
    prime_flag = True

    for i in range(2, int(sqrt(n)) + 1):
        if (n % i) == 0:
            prime_flag = False
            break

    if prime_flag:
        l.append(n)
        count = count + 1
    n = n + 1
print(l)
print('sum of first ten prime numbers: ',sum(l))
```

```
Enter a number: 10
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[165], line 23
     21     n = n + 1
     22 print(l)
---> 23 print('sum of first ten prime numbers: ',sum(l))

TypeError: 'int' object is not callable
```

In [166]:

```
#8 Write a python program to implement arithematic operations using nested if statement

S1= int (input("Enter the 1st number: "))
S2= int(input("Enter the 2nd number: "))

opr= input("Enter the Operation: ")

if opr == 'Addition':
    print(S1+S2)
elif opr == 'Subtraction':
    print(S1 - S2)
elif opr == 'Multiplication':
    print(S1*S2)

else:
    print(S1/S2)
```

```
Enter the 1st number: 20
Enter the 2nd number: 10
Enter the Operation: Subtraction
10
```

In [167]:

```
#9 Write a python program to take the temperature in celsius and convert it to a fahrenhit

c=float(input("Enter the Celsius Value: "))

f= (c * 1.8) +32

print('{} Celsius equals to {} Fahrenhit'.format(c,f))
```

```
Enter the Celsius Value: 37
37.0 Celsius equals to 98.60000000000001 Fahrenhit
```

In [170]:

```
#10 Write a python function to find the minimum and maximum number in List without using any inbuilt function

List=[27,873,9773,987,3552,987]

max=0
min=List[0]
for i in List:
    if i>max:
        max=i
    if i<min:
        min=i

print("The maximum number is: ", max)
print("The manimum number is: ", min)
```

```
The maximum number is:  9773
The manimum number is:  27
```

In [175]:

```
#11 Write a program in python to print out the number of seconds in 30 day momth 30 days,24 hrs in a day,60 minutes per day, 60 s

Sec=[]
Month=30
Day=24
Hour=60
Minute=60

Sec_month=Month*Day*Hour*Minute
print(Sec_month)

Sec_Day=Day*Hour*Minute
print (Sec_Day)

Sec_Hour=Hour*Minute
print(Sec_Hour)

Sec_Minute=1*Minute
print(Sec_Minute)
```

```
2592000
86400
3600
60
```

In [178]:

```
#12 Write a program in python to print out the number of seconds in year


Days=365
Hour=24
Minute=60
Sec=60

Sec_Year=Days*Hour*Minute*Sec

print(Sec_Year)
```

```
31536000
```

In [181]:

```
 speed of 150mph, how long will it take a train travelling at this speed to travel from London to Glasgow which is 414 miles away
```

```
2.76
```

In [182]:

```
#14  Write a python program that defines a vraibele and assigns 192 days to the variable. the program should then print the total

days_in_each_school=192
thrs=0
for i in range(7,12):
    thrs+=(192*6)
print("Total hours: ", thrs)
```

```
Total hours:  5760
```

In [188]:

```
#15 if the age of ram,sam,khan are input through the keyboard, write a python program to determine the eldets and youngest pof th

r= int(input("Enter the Age of Ram :"))
s= int(input("Enter the Age of Sam :"))
k = int(input("Enter the Age of khan :"))

if(r < s and r <k):
    print("The Youngest Age is Ram")

elif(s<r and s<k):
    print("The Youngest Age is Sam")

else:
    print("The Youngest Age is Khan")

if(r > s and r >k):
    print("The Eldest Age is Ram")

elif(s>r and s>k):
    print("The Edlest Age is Sam")

else:
    print("The Edlest Age is Khan")
```

```
Enter the Age of Ram :25
Enter the Age of Sam :52
Enter the Age of khan :12
The Youngest Age is Khan
The Edlest Age is Sam
```

In [ ]: