Python imp

```
alpha='abcdefghijklmnopqrstuvwxyz'

s='chennai'
#I expect to output tvebstibo

t=''

i=0
t=t+(alpha[(((alpha.index(s[i]))+1)%26)] )
t=t+(alpha[(((alpha.index(s[i+1]))+1)%26)])
t=t+(alpha[(((alpha.index(s[i+2]))+1)%26)])
t=t+(alpha[(((alpha.index(s[i+3]))+1)%26)])
t=t+(alpha[(((alpha.index(s[i+4]))+1)%26)] )
t=t+(alpha[(((alpha.index(s[i+5]))+1)%26)] )
t=t+(alpha[(((alpha.index(s[i+6]))+1)%26)] )

print(t)
```

```
Python 3.8.1 (default, Jan  8 2020, 16:15:59)
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

Restarting kernel...



In [1]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
difoo

In [2]: runfile('/Users/srsiyengar/.spyder-py3/pod.py', wdir='/Users/
srsiyengar/.spyder-py3')
difoobj

In [3]:
```

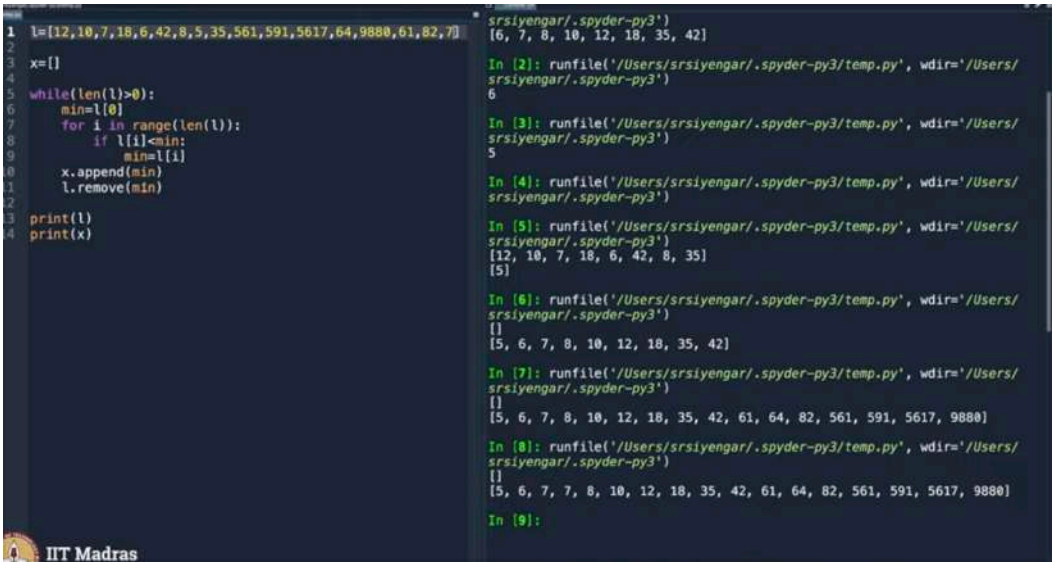
```
import random

l=[]
#create an empty list.

for i in range(75):
    l.append(random.randint(1,365))
    #append random numbers between 1 to 365.
    #append 30 of them
l.sort()
print(l)
i=0
flag=0 #denotes that there is no repetition
while(i<len(l)-1):
    if (l[i]==l[i+1]):
        print("Repeats",l[i],l[i+1])
        flag=1
        break
    i=i+1

if (flag==0):
    print("There is no repetition.")
```

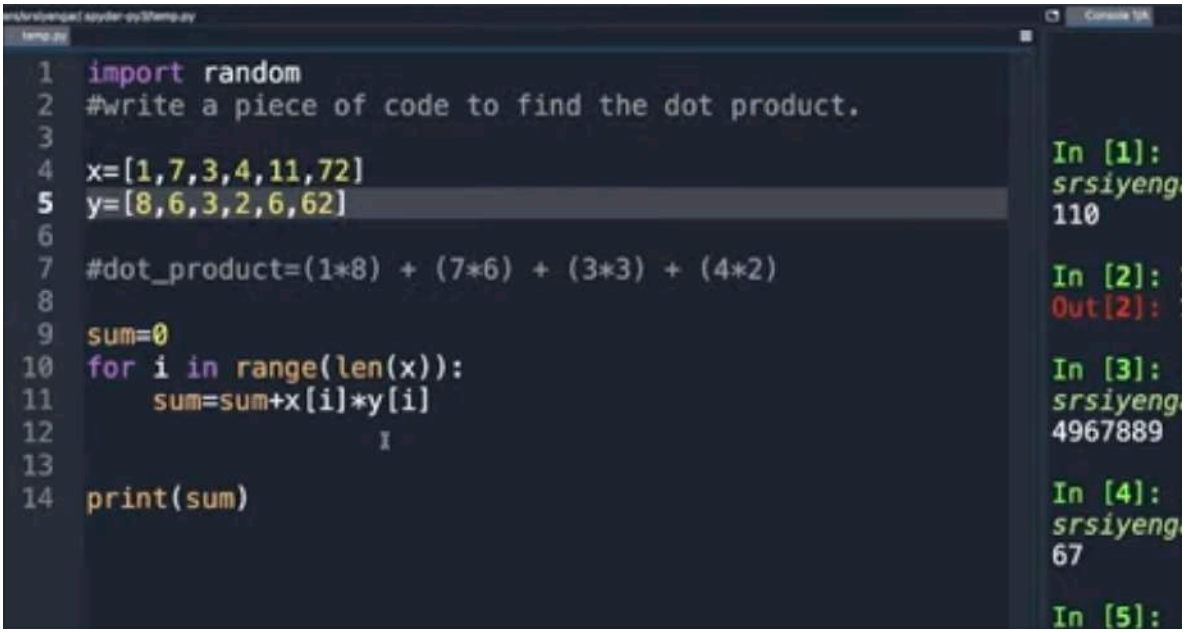
```
174, 182, 184, 188, 191, 196, 198, 204, 206, 209, 216, 220, 222, 243, 249,
256, 258, 264, 265, 272, 302, 304, 308, 312, 315, 318, 319, 328, 329, 330,
349, 352, 356]
Repeats 10 10

In [140]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[9, 16, 24, 24, 31, 32, 32, 34, 40, 40, 48, 48, 48, 53, 59, 62, 71, 72, 85
103, 104, 104, 110, 112, 114, 118, 123, 124, 128, 134, 144, 146, 150, 168,
194, 195, 207, 217, 219, 219, 223, 224, 226, 226, 235, 236, 240, 241, 244,
269, 271, 271, 273, 283, 284, 300, 304, 310, 310, 314, 324, 326, 329, 335,
353, 354, 360]
Repeats 24 24

In [141]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[1, 4, 7, 15, 19, 23, 25, 27, 27, 35, 41, 46, 47, 47, 48, 49, 50, 63, 67,
103, 111, 114, 120, 121, 121, 130, 130, 135, 140, 149, 164, 167, 170, 179,
197, 201, 207, 212, 217, 218, 221, 229, 234, 234, 234, 234, 235, 241, 243,
258, 261, 262, 268, 270, 276, 281, 284, 292, 295, 300, 302, 320, 320, 335,
341, 361]
Repeats 27 27

In [142]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[5, 6, 8, 10, 14, 15, 22, 27, 32, 33, 39, 48, 51, 65, 73, 80, 92, 99, 105,
130, 140, 145, 159, 165, 166, 169, 170, 189, 192, 195, 198, 199, 202, 202,
223, 229, 232, 237, 237, 238, 243, 253, 254, 256, 257, 258, 258, 272, 285,
303, 307, 308, 314, 324, 325, 327, 333, 334, 338, 341, 342, 343, 343, 343,
350, 353, 365]
Repeats 202 202

In [143]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[5, 19, 20, 33, 41, 52, 53, 59, 61, 78, 84, 89, 94, 95, 105, 105, 106, 114
119, 119, 120, 128, 132, 136, 143, 151, 154, 160, 169, 171, 177, 183, 190,
201, 201, 205, 209, 210, 218, 228, 229, 231, 231, 231, 240, 248, 259, 260,
272, 272, 283, 284, 288, 290, 292, 292, 299, 299, 307, 308, 310, 311, 323,
353, 359, 364, 364]
```

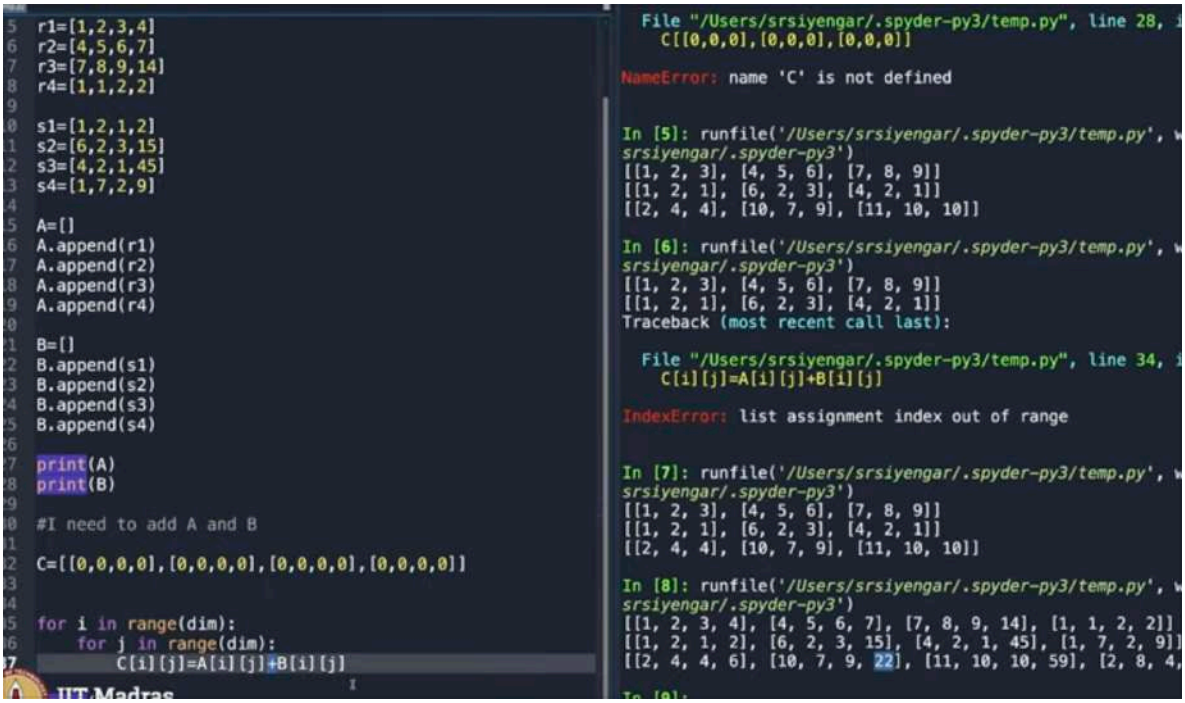
```
import random

l=[]

#for i in range(1000000):
#    l.append(random.randint(1,10000000))


l=[2001,1990,1981,1985,2003,1988,1999]

n=0
while(n>-1):
    print("Enter a number, type -1 to exit:")
    n=int(input())

    flag=0

    for i in range(len(l)):
        if (n==l[i]):
            print("Hip Hip Hurray, element found")
            flag=1
            break;
    if (flag==0):
        print("Element not found")
```

```
138947
Element not found
Enter a number, type -1 to exit:

123421
Element not found
Enter a number, type -1 to exit:

897234
Element not found
Enter a number, type -1 to exit:

867234
Element not found
Enter a number, type -1 to exit:

78678
Element not found
Enter a number, type -1 to exit:

-1
Element not found

In [23]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
Enter a number, type -1 to exit:

2001
Hip Hip Hurray, element found
Enter a number, type -1 to exit:

1981
Hip Hip Hurray, element found
Enter a number, type -1 to exit:

2000
Element not found
Enter a number, type -1 to exit:
```

IIT Madras

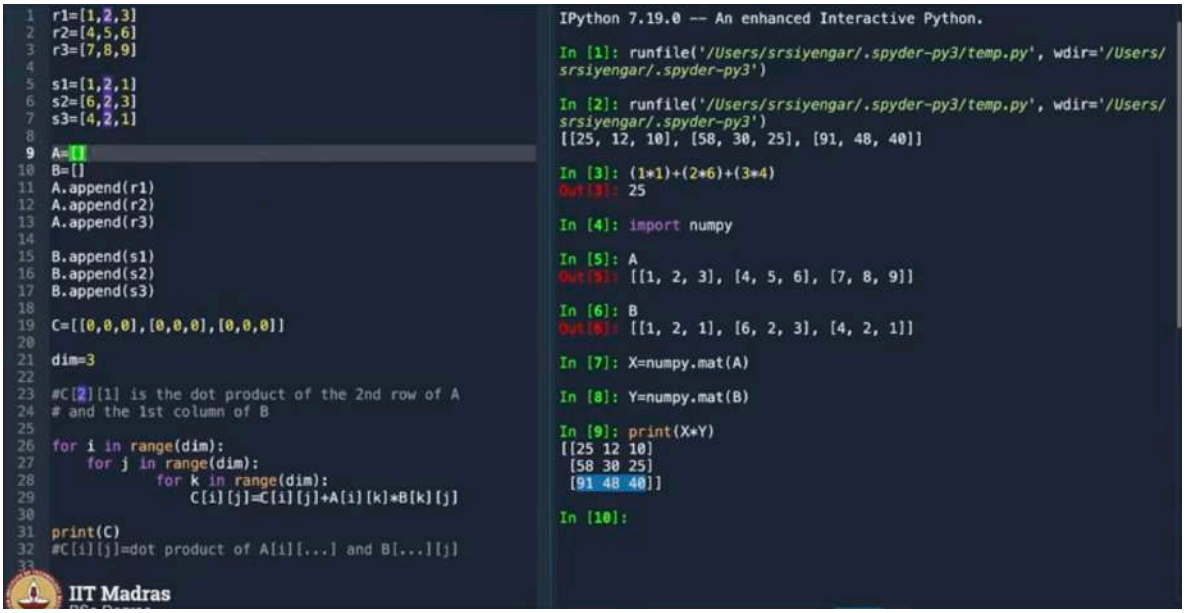

l=[12,10,7,18,6,42,8,5,35,561,591,5617,64,9880,61,82,7]
x=[]
while(len(l)>0):
min=l[0]
for i in range(len(l)):
if l[i]<min:
min=l[i]
x.append(min)
l.remove(min)
print(l)
print(x)
srsiyengar/.spyder-py3')
[6, 7, 8, 10, 12, 18, 35, 42]
In [2]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
6
In [3]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
5
In [4]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
[12, 10, 7, 18, 6, 42, 8, 35]
[5]
In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
[]
[5, 6, 7, 8, 10, 12, 18, 35, 42]
In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
[]
[5, 6, 7, 8, 10, 12, 18, 35, 42, 61, 64, 82, 561, 591, 5617, 9880]
In [8]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/srsiyengar/.spyder-py3')
[]
[5, 6, 7, 7, 8, 10, 12, 18, 35, 42, 61, 64, 82, 561, 591, 5617, 9880]
In [9]:
IIT Madras


import random
#write a piece of code to find the dot product.
x=[1,7,3,4,11,72]
y=[8,6,3,2,6,62]
#dot_product=(1*8) + (7*6) + (3*3) + (4*2)
sum=0
for i in range(len(x)):
sum=sum+x[i]*y[i]
print(sum)
In [1]:
110
In [2]:
Out[2]:
In [3]:
4967889
In [4]:
67
In [5]:



```
r1=[1,2,3,4]
r2=[4,5,6,7]
r3=[7,8,9,14]
r4=[1,1,2,2]

s1=[1,2,1,2]
s2=[6,2,3,15]
s3=[4,2,1,45]
s4=[1,7,2,9]

A=[]
A.append(r1)
A.append(r2)
A.append(r3)
A.append(r4)

B=[]
B.append(s1)
B.append(s2)
B.append(s3)
B.append(s4)

print(A)
print(B)

#I need to add A and B

C=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]


for i in range(dim):
    for j in range(dim):
        C[i][j]=A[i][j]+B[i][j]
```

```
  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28, i
    C[[0,0,0],[0,0,0],[0,0,0]]

NameError: name 'C' is not defined


In [5]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', w
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [6]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', w
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
Traceback (most recent call last):

  File "/Users/srsiyengar/.spyder-py3/temp.py", line 34, i
    C[i][j]=A[i][j]+B[i][j]

IndexError: list assignment index out of range


In [7]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', w
srsiyengar/.spyder-py3')
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
[[1, 2, 1], [6, 2, 3], [4, 2, 1]]
[[2, 4, 4], [10, 7, 9], [11, 10, 10]]

In [8]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', w
srsiyengar/.spyder-py3')
[[1, 2, 3, 4], [4, 5, 6, 7], [7, 8, 9, 14], [1, 1, 2, 2]]
[[1, 2, 1, 2], [6, 2, 3, 15], [4, 2, 1, 45], [1, 7, 2, 9]]
[[2, 4, 4, 6], [10, 7, 9, 22], [11, 10, 10, 59], [2, 8, 4,

In [9]:
```



```
r1=[1,2,3]
r2=[4,5,6]
r3=[7,8,9]

s1=[1,2,1]
s2=[6,2,3]
s3=[4,2,1]

A=[]
B=[]
A.append(r1)
A.append(r2)
A.append(r3)

B.append(s1)
B.append(s2)
B.append(s3)

C=[[0,0,0],[0,0,0],[0,0,0]]

dim=3

#C[2][1] is the dot product of the 2nd row of A
# and the 1st column of B

for i in range(dim):
    for j in range(dim):
            for k in range(dim):
                C[i][j]=C[i][j]+A[i][k]*B[k][j]

print(C)
#C[i][j]=dot product of A[i][...] and B[...][j]
```

```
IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')

In [2]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/Users/
srsiyengar/.spyder-py3')
[[25, 12, 10], [58, 30, 25], [91, 48, 40]]

In [3]: (1*1)+(2*6)+(3*4)
Out[3]: 25

In [4]: import numpy

In [5]: A
Out[5]: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

In [6]: B
Out[6]: [[1, 2, 1], [6, 2, 3], [4, 2, 1]]

In [7]: X=numpy.mat(A)

In [8]: Y=numpy.mat(B)

In [9]: print(X*Y)
[[25 12 10]
 [58 30 25]
 [91 48 40]]

In [10]:
```

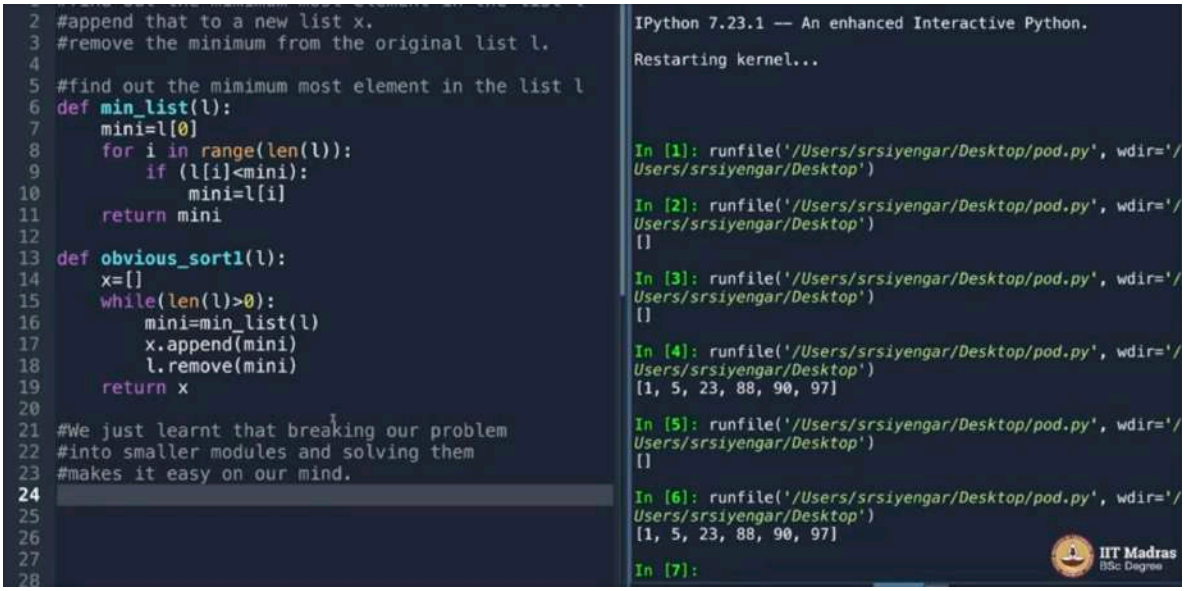
FUNCTIONS

```
def list_appendbefore(l,z):
    newl=[]
    for i in range(len(z)):
        newl.append(z[i])
    for i in range(len(l)):
        newl.append(l[i])
    return newl

def list_appendend(l,z):
    newl=[]
    for i in range(len(l)):
        newl.append(l[i])
    for i in range(len(z)):
        newl.append(z[i])
    return newl

def list_average(l):
    sum=0
    fori
```

IIT Madras
BSc Degree

```
In [7]: l=[1,2,7,8,9]

In [8]: z=[7,51,1]

In [9]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/
Users/srsiyengar/.spyder-py3')
-10
100
[10, 20, 30, 1, 2, 3, 4, 5, -10, 6, 4, 100]

In [10]: runcell(0, '/Users/srsiyengar/.spyder-py3/temp.py')
  File "/Users/srsiyengar/.spyder-py3/temp.py", line 28
    l=[1,2,3,4,5,-10,6,4,100]
    ^
IndentationError: expected an indented block


In [11]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/
Users/srsiyengar/.spyder-py3')
-10
100
[10, 20, 30, 1, 2, 3, 4, 5, -10, 6, 4, 100]

In [12]: runfile('/Users/srsiyengar/.spyder-py3/temp.py', wdir='/
Users/srsiyengar/.spyder-py3')
-10
100
[10, 20, 30, 1, 2, 3, 4, 5, -10, 6, 4, 100]
[1, 2, 3, 4, 5, -10, 6, 4, 100, 10, 20, 30]

In [13]:
```

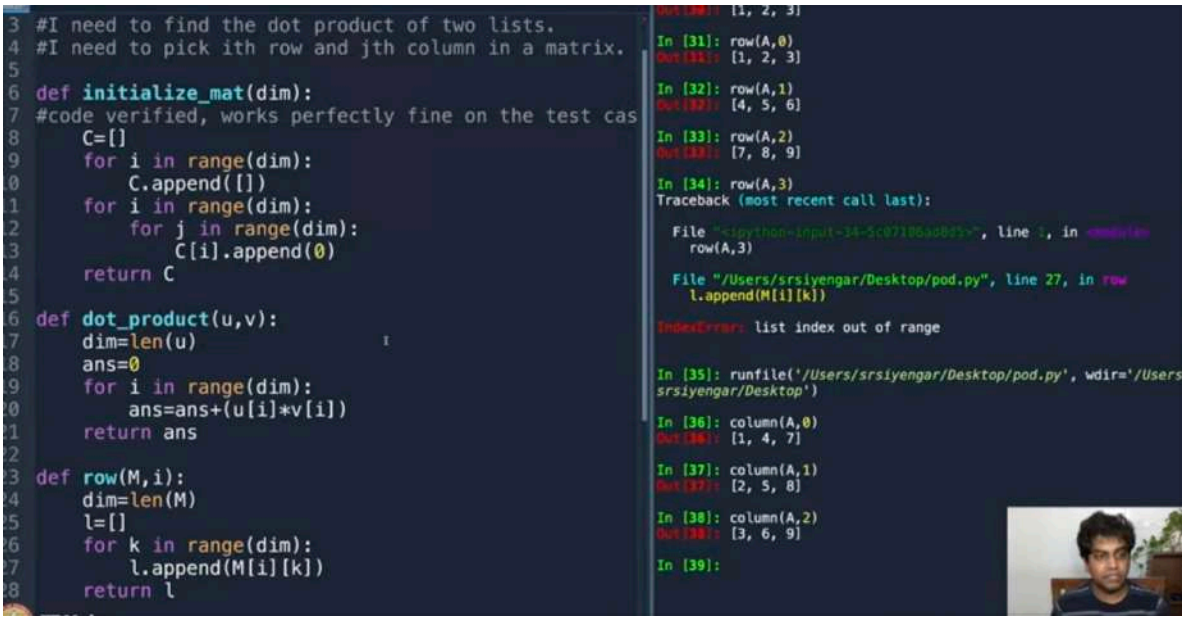
```
def list_min(l):
    mini=l[0]
    for i in range(len(l)):
        if (l[i]<mini):
            mini=l[i]
    return mini

def list_maxi(l):
    maxi=l[0]
    for i in range(len(l)):
        if (l[i]>maxi):
            maxi=l[i]
    return maxi

def list_appendbefore(l,z):
    newl=[]
    for i in range(len(z)):
        newl.append(z[i])
    for i in range(len(l)):
        newl.append(l[i])
    return newl

def list_appendend(l,z):
    newl=[]
    for i in range(len(l)):
        newl.append(l[i])
    for i in range(len(z)):
```



```
#append that to a new list x.
#remove the minimum from the original list l.

#find out the mimimum most element in the list l
def min_list(l):
    mini=l[0]
    for i in range(len(l)):
        if (l[i]<mini):
            mini=l[i]
    return mini

def obvious_sort1(l):
    x=[]
    while(len(l)>0):
        mini=min_list(l)
        x.append(mini)
        l.remove(mini)
    return x

#We just learnt that breaking our problem
#into smaller modules and solving them
#makes it easy on our mind.
```

```
IPython 7.23.1 -- An enhanced Interactive Python.

Restarting kernel...

In [1]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')

In [2]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [3]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [4]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [5]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[]

In [6]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/
Users/srsiyengar/Desktop')
[1, 5, 23, 88, 90, 97]

In [7]:
```

IIT Madras BSc Degree



```
#I need to find the dot product of two lists.
#I need to pick ith row and jth column in a matrix.

def initialize_mat(dim):
#code verified, works perfectly fine on the test cas
    C=[]
    for i in range(dim):
        C.append([])
    for i in range(dim):
        for j in range(dim):
            C[i].append(0)
    return C

def dot_product(u,v):
    dim=len(u)
    ans=0
    for i in range(dim):
        ans=ans+(u[i]*v[i])
    return ans

def row(M,i):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[i][k])
    return l
```

```
Out[30]: [1, 2, 3]

In [31]: row(A,0)
Out[31]: [1, 2, 3]

In [32]: row(A,1)
Out[32]: [4, 5, 6]

In [33]: row(A,2)
Out[33]: [7, 8, 9]

In [34]: row(A,3)
Traceback (most recent call last):

  File "<ipython-input-34-5c07106a0805>", line 1, in <module>
    row(A,3)

  File "/Users/srsiyengar/Desktop/pod.py", line 27, in row
    l.append(M[i][k])

IndexError: list index out of range


In [35]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users
srsiyengar/Desktop')

In [36]: column(A,0)
Out[36]: [1, 4, 7]

In [37]: column(A,1)
Out[37]: [2, 5, 8]

In [38]: column(A,2)
Out[38]: [3, 6, 9]

In [39]:
```

```python
def row(M,i):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[i][k])
    return l

def column(M,j):
    dim=len(M)
    l=[]
    for k in range(dim):
        l.append(M[k][j])
    return l

def mat_mul(A,B):
    dim=len(A)
    C=initialize_mat(dim)
    for i in range(dim):
        for j in range(dim):
            C[i][j]= dot_product(row(A,i),column(B,j
    return C
```

IIT Madras
BSc Degree

```
SyntaxError: invalid syntax

In [41]: runfile('/Users/srsiyengar/Desktop/pod.py', wdir='/Users/
srsiyengar/Desktop')

In [42]: A
Out[42]: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

In [43]: B=[[1,2,1],[3,1,7],[6,2,3]]

In [44]: mat_mul(A,B)
Out[44]: [[25, 10, 24], [55, 25, 57], [85, 40, 90]]

In [45]: (1*1)+(2*3)+(3*6)
Out[45]: 25

In [46]: 4 5 6 with 1 7 3
  File "<ipython-input-46-3197c99d040a>", line 1
    4 5 6 with 1 7 3
      ^
SyntaxError: invalid syntax

In [47]: (1*1)+(5*7)+(6*3)
Out[47]: 54

In [48]: (4*1)+(5*7)+(6*3)
Out[48]: 57

In [49]: import numpy

In [50]: A=
```

$$f(n) = f(n-1) \cdot (1.1)$$

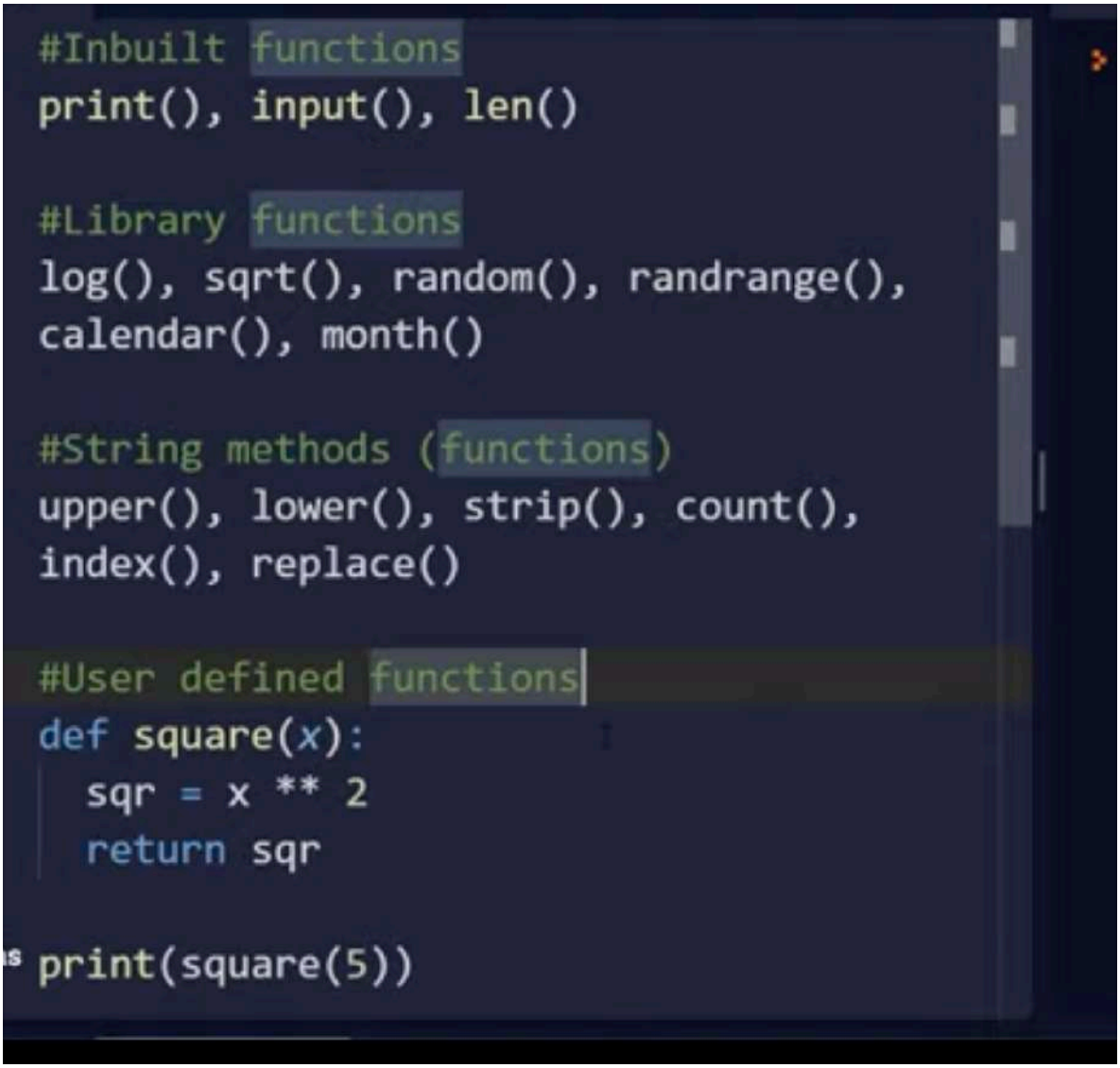
$$Sum(n) = Sum(n-1) + n$$

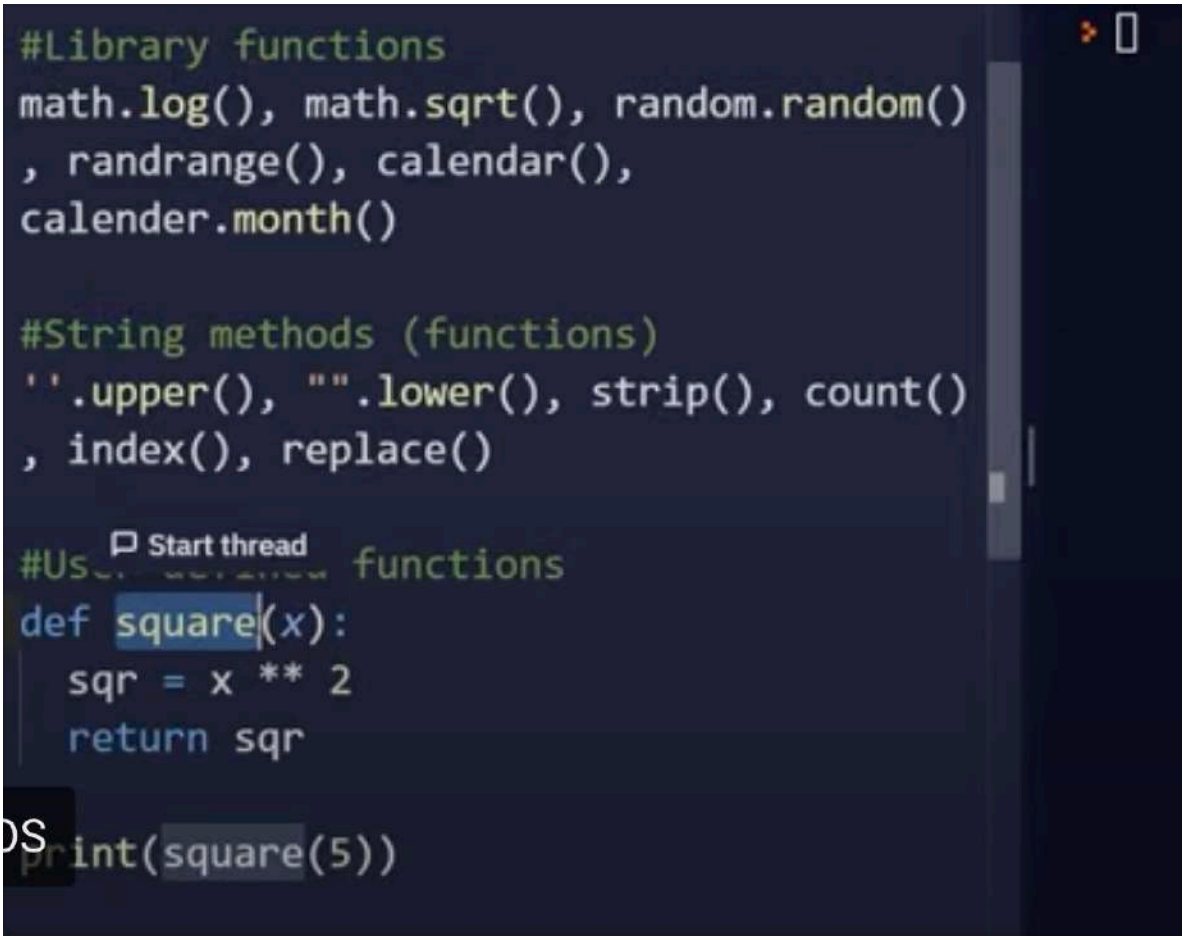
$$Fact(n) = [Fact(n-1)] \cdot n$$

IIT Madras
BSc Degree

```python
#recursion in python

def sum(n):
#verified
    if (n==1):
        return 1
    else:
        return n+sum(n-1)

#python lets you call the same function within the function.


#compute compound interest by assuming the interest to be 10%
def comp(p,n):
#verified
    if (n==1):
        return p*(1.1)
    else:
        return (comp(p,n-1))*1.1

def fact(n):
    if (n==1):
        return 1
    else:
        return (fact(n-1))*n

print(fact(5))
```

```
Out[16]: 2662.0

In [17]: runfile('/Users/srsiyengar/Desktop/pod.py',
wdir='/Users/srsiyengar/Desktop')

In [18]: runfile('/Users/srsiyengar/Desktop/pod.py',
wdir='/Users/srsiyengar/Desktop')
120

In [19]: 5*4*3*2*`
  File "<ipython-input-19-989c6ae6ac47>", line 1
    5*4*3*2*`
      ^
SyntaxError: invalid syntax

In [20]: 5*4*3*2*1
Out[20]: 120

In [21]:
```


#Inbuilt functions
print(), input(), len()
#Library functions
log(), sqrt(), random(), randrange(),
calendar(), month()
#String methods (functions)
upper(), lower(), strip(), count(),
index(), replace()
#User defined functions
def square(x):
sqr = x ** 2
return sqr
print(square(5))



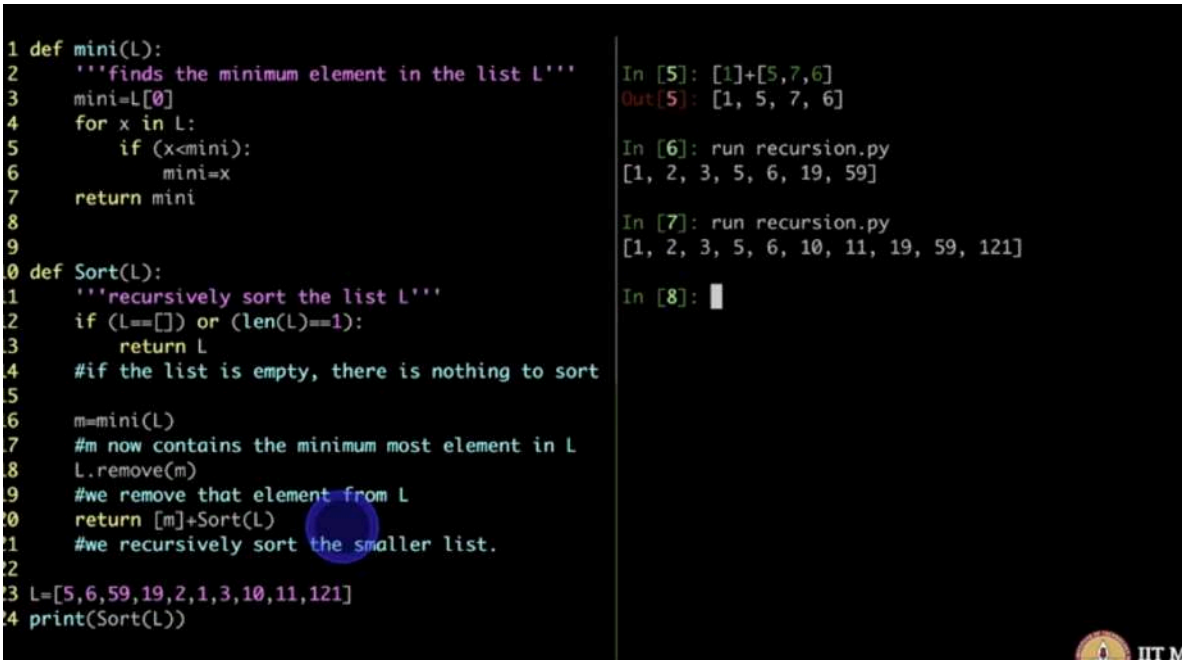
Property	List	Tuple	Dictionary	Set
Notation	[ ]	( )	{'Key': 'Value'}	{ }
Creation	list()	tuple()	dict()	set()
Mutability	Mutable	Immutable	Mutable	Mutable
Type of elements which can be stored	Any	Any	Keys: Hashable Values: Any	Hashable
Order of elements	Ordered	Ordered	Unordered*	Unordered
Duplicate elements	Allowed	Allowed	Keys: Not allowed Values: Allowed	Not allowed
Operations	Add, Update, Delete	None	Keys: Add, Delete Values: Add, Update, Delete	Add, Delete
Operations	Indexing, Slicing, Iteration	Indexing, Slicing, Iteration	Iteration	Iteration
Sorting	Possible	Not possible	Possible	Not possible

*Python 3.6 and earlier. Dictionaries are ordered as per Python 3.7 and above.

List methods	Tuple methods	Dictionary methods	Set methods
append() clear() copy() count() extend() index() insert() pop() remove() reverse() sort()	count() index()	clear() copy() fromkeys() get() items() keys() pop() popitem() setdefault() update() values()	add() clear() copy() difference() difference_update() discard() intersection() intersection_update() isdisjoint() issubset() issuperset() pop() remove() symmetric_difference() symmetric_difference_update() union() update()

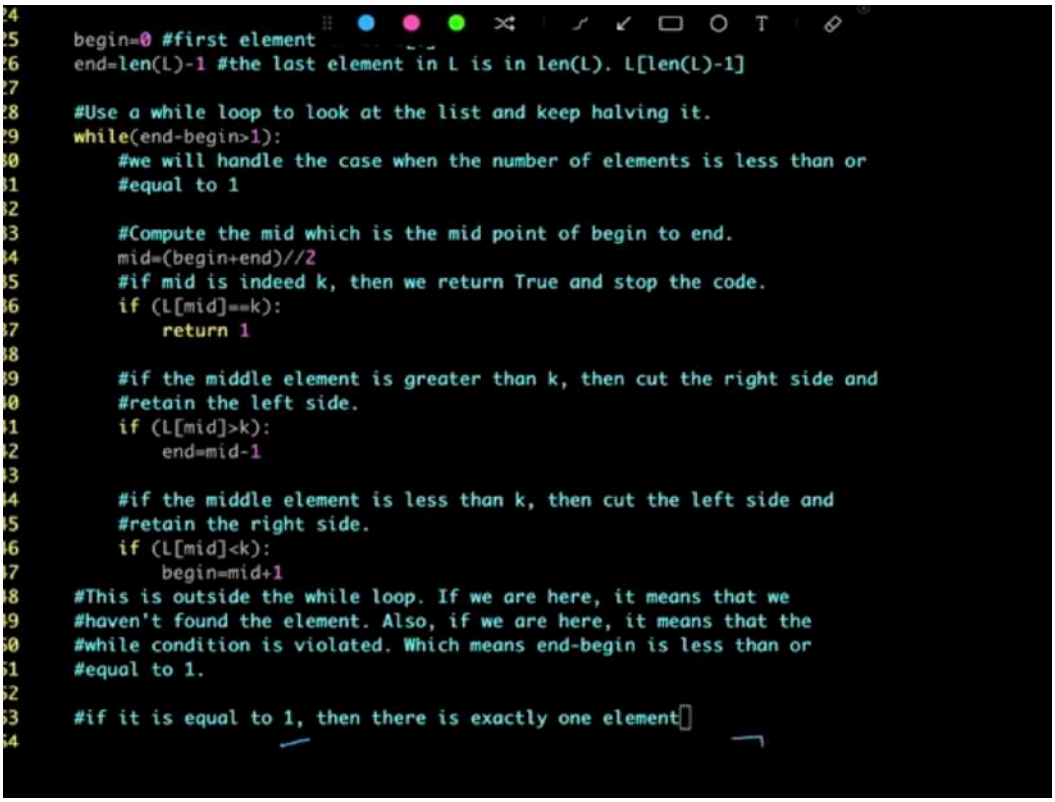
```python
stats = dict()

for i in range(4):
    L = input().split(',')
    # 'Virat,101,88,93,0,120' -> ['Virat', '101', '88', '93', '0', '120']
    name = L[0]
    total = 0

    for runs in L[1:]:
        total = total + int(runs)

    stats[name] = total
```

```python
def is_long_tail(x):
    num = str(x)
    first, second = num.split('.')
    return len(second) > len(first)

def long_tail(L):
    count = 0
    for x in L:
        if is_long_tail(x):
            count = count + 1
    return count
```



```
def mini(L):
    '''finds the minimum element in the list L'''
    mini=L[0]
    for x in L:
        if (x<mini):
            mini=x
    return mini


def Sort(L):
    '''recursively sort the list L'''
    if (L==[]) or (len(L)==1):
        return L
    #if the list is empty, there is nothing to sort

    m=mini(L)
    #m now contains the minimum most element in L
    L.remove(m)
    #we remove that element from L
    return [m]+Sort(L)
    #we recursively sort the smaller list.

L=[5,6,59,19,2,1,3,10,11,121]
print(Sort(L))
```

```
In [5]: [1]+[5,7,6]
Out[5]: [1, 5, 7, 6]

In [6]: run recursion.py
[1, 2, 3, 5, 6, 19, 59]

In [7]: run recursion.py
[1, 2, 3, 5, 6, 10, 11, 19, 59, 121]

In [8]:
```

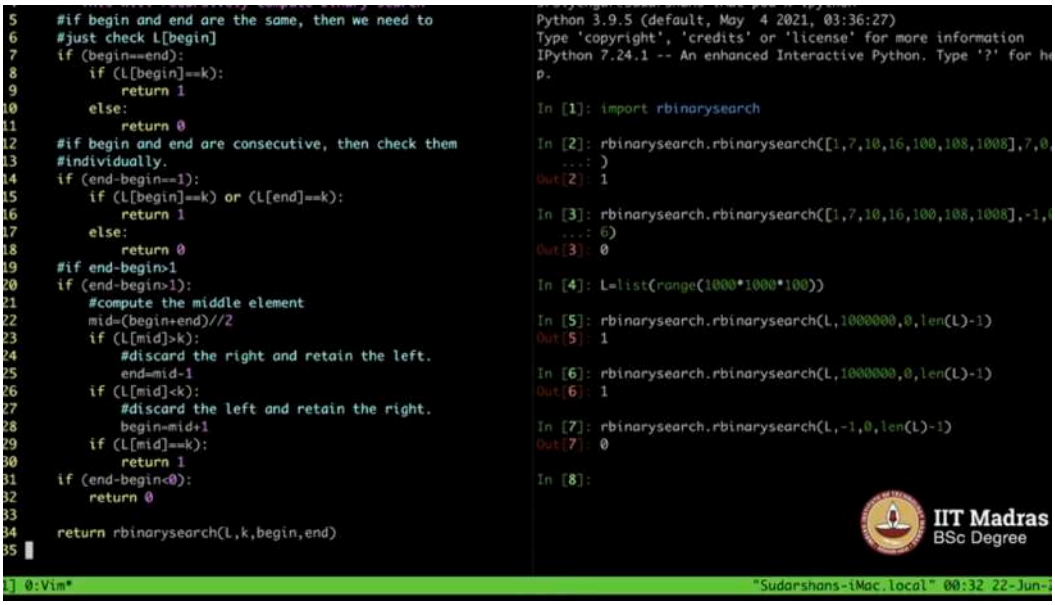


```
    begin=0 #first element
    end=len(L)-1 #the last element in L is in len(L). L[len(L)-1]

    #Use a while loop to look at the list and keep halving it.
    while(end-begin>1):
        #we will handle the case when the number of elements is less than or
        #equal to 1

        #Compute the mid which is the mid point of begin to end.
        mid=(begin+end)//2
        #if mid is indeed k, then we return True and stop the code.
        if (L[mid]==k):
            return 1

        #if the middle element is greater than k, then cut the right side and
        #retain the left side.
        if (L[mid]>k):
            end=mid-1

        #if the middle element is less than k, then cut the left side and
        #retain the right side.
        if (L[mid]<k):
            begin=mid+1
    #This is outside the while loop. If we are here, it means that we
    #haven't found the element. Also, if we are here, it means that the
    #while condition is violated. Which means end-begin is less than or
    #equal to 1.

    #if it is equal to 1, then there is exactly one element
```

```
#We want to shrink my list
#We will do that using a while loop.

begin=0 #first element in L. L[0]
end=len(L)-1 #the last element in L is in len(L). L[len(L)-1]

#Use a while loop to look at the list and keep halving it.
while(end-begin>1):
    #we will handle the case when the number of elements is less than or
    #equal to 1

    #Compute the mid which is the mid point of begin to end.
    mid=(begin+end)//2
    #if mid is indeed k, then we return True and stop the code.
    if (L[mid]==k):
        return 1

    #if the middle element is greater than k, then cut the right side and
    #retain the left side.
    if (L[mid]>k):
        end=mid-1

    #if the middle element is less than k, then cut the left side and
    #retain the right side.
    if (L[mid]<k):
        begin=mid+1
#This is outside the while loop. If we are here, it means that we
#haven't found the element. Also, if we are here, it means that the
#while condition is violated. Which means end-begin is less than or
#equal to 1.

#if it is equal to 1, then there is exactly two elements
if (L[begin]==k) or (L[end]==k):
    return 1
else:
    return 0
```

```python

def obvious_search(L,k):
    '''Check if a given element k
    is present in a list L or not. This function
    was authored by S. R. S. Iyengar'''
    for x in L:
        if x==k:
            return 1
    return 0
    #code verified. Working fine.

'''A question: Can we write a piece of code that
searches for a given element in the list L faster than the obvious algorithm
given above :-( :-( :-('''

def binary_search(L,k):
    '''This function is an alternative for the obvious search. It
    does exactly what is expected from the obvious_search, but in
    an efficient way. This method is popularly called the
    binary search.'''

    #We want to shrink my list
    #We will do that using a while loop.

    begin=0 #first element in L. L[0]
    end=len(L)-1 #the last element in L is in len(L). L[len(L)-1]

    #Use a while loop to look at the list and keep halving it.
    while(end-begin>1):
        #we will handle the case when the number of elements is less than or
        #equal to 1

        #Compute the mid which is the mid point of begin to end.
        mid=(begin+end)//2
        #if mid is indeed k, then we return True and stop the code.
        if (L[mid]==k):
            return 1

        #if the middle element is greater than k, then cut the right side and
        #retain the left side.
```

```
def rbinarysearch(L,k,begin,end):
    '''This will recursively compute binary search'''
    #if begin and end are the same, then we need to
    #just check L[begin]
    if (begin==end):
        if (L[begin]==k):
            return 1
        else:
            return 0
    #if begin and end are consecutive, then check them
    #individually.
    if (end-begin==1):
        if (L[begin]==k) or (L[end]==k):
            return 1
        else:
            return 0
    #if end-begin>1
    if (end-begin>1):
        #compute the middle element
        mid=(begin+end)//2
        if (L[mid]>k):
            #discard the right and retain the left
            end=mid-1
        if (L[mid]<k):
            #discard the left and retain the right.
```

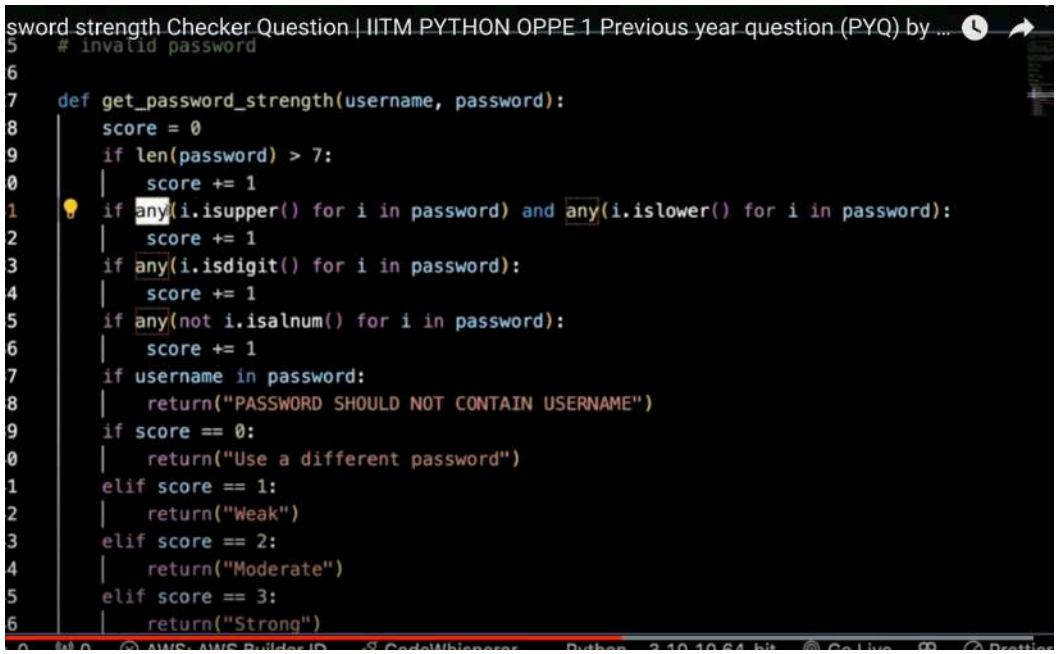



```
date = input()
# mm/dd/yyyy

mm, dd, yy = date[:2], date[3:5], date[-2:]

print(dd, mm, yy, sep = '-')
```

```
11 def twin_primes(p, q):
12     '''
13     Determine if p and q are twin primes
14
15     Parameters:
16         p: int
17         q: int
18     Return:
19         result: bool
20     '''
21     if abs(p - q) == 2:
22         return is_prime(p) and is_prime(q)
23     return False
24
25
```

```
def twin_primes(p, q):
    '''
    Determine if p and q are twin primes

    Parameters:
        p: int
        q: int
    Return:
        result: bool
    '''
    if is_prime(p) and is_prime(q):
        return abs(p - q) == 2
    return False
```

```
36          score += 1
37      if username in password:
38          return("PASSWORD SHOULD NOT CONTAIN USERNAME")
39      if score == 0:
40          return("Use a different password")
41      elif score == 1:
42          return("Weak")
43      elif score == 2:
44          return("Moderate")
45      elif score == 3:
46          return("Strong")
47      elif score == 4:
48          return("Very Strong")
49
```



```python
def rotate(L):
    L.insert(0, L.pop())
    return L


L = input().split(",")
k = int(input("Enter a positive integer: "))
for i in range(k):
    L = rotate(L)
print(L)
```

```python
def evaluate(s):
    temp_dict = {'zero': 0, 'one': 1, 'two': 2, 'three': 3, 'four': 4,'five':
    'six': 6, 'seven': 7, 'eight': 8, 'nine': 9}
    s = s.split() # split the string into a list
    ans = 0
    for i in range(0,len(s),2):
        # if starts with plus or minus
        if s[i] == 'plus':
            ans += temp_dict[s[i+1]]
        elif s[i] == 'minus':
            ans -= temp_dict[s[i+1]]
        else:
            # it will be a number
            if i == 0:
                ans = temp_dict[s[i]]
            else:
                if s[i-1] == 'plus':
                    ans += temp_dict[s[i]]
                elif s[i-1] == 'minus':
                    ans -= temp_dict[s[i]]
    return ans
s = input("Enter a sequence of space-separated words: ")
```

```
]: def valid_phone_number(phone_number):
    if len(phone_number) != 10:
        return False
    if phone_number[:5] != '98123':
        return False
    for i in phone_number:
        if phone_number.count(i) > 5:
            return False
    return True


D = {}
while True:
    phone_number = input()
    if phone_number == 'STOP':
        break
    if valid_phone_number(phone_number):
        D[phone_number] = 'VALID'
    else:
        D[phone_number] = 'INVALID'
print(D)
```

```
'''This program considers an input file and encrypts it by using caesar
cipher. By that we mean, we shift the letters by 3 units. For example,
a becomes d, b becomes e and so on... w becomes z, x becomes a,
y becomes b and z becomes c'''

import string


def create_caesar_dictionary():
    l=string.ascii_lowercase
    l=list(l)
    d={}
    for i in range(len(l)):
        d[l[i]]=l[(i+3)%26]
    return d


f=open('sherlock.txt','r')
g=open('encrypted_sherlock.txt','w')
d=create_caesar_dictionary()

c=f.read(1)
while (c!=''):
    g.write(d[c])
    c=f.read(1)

f.close()
g.close()


"caesar.py" 36L, 596B written
```

```
In [15]: l=list(l)

In [16]: d={}

In [17]: for i in range(len(l)):
    ...:     d[l[i]]=l[(i+3)%26]
    ...:

In [18]: print(d)
{'a': 'd', 'b': 'e', 'c': 'f', 'd': 'g', 'e': 'h'
, 'f': 'i', 'g': 'j', 'h': 'k', 'i': 'l', 'j': 'm
', 'k': 'n', 'l': 'o', 'm': 'p', 'n': 'q', 'o': '
r', 'p': 's', 'q': 't', 'r': 'u', 's': 'v', 't':
'w', 'u': 'x', 'v': 'y', 'w': 'z', 'x': 'a', 'y':
 'b', 'z': 'c'}

In [19]: d['a']
Out[19]: 'd'

In [20]: d['z']
Out[20]: 'c'

In [21]: d['y']
Out[21]: 'b'

In [22]: d['w']
Out[22]: 'z'

In [23]: run caesar.py

In [24]:
```

```
In [17]: x=open('mytext.txt','r')

In [18]: s=x.read()

In [19]: print(s)
Sudarshan yourname IIT C++ Germany

In [20]: x=open('mytext.txt','r')

In [21]: s=x.read()

In [22]: print(s)
Sudarshan
Ajit
Amit
Tatsavit
Lakshmi
Bhawana
Ramya
Uday

In [23]:
```

```python
class Student:
    roll_no = None
    name = None

s0 = Student()
s0.roll_no = 0
s0.name = 'Bhuvanesh'
print(s0.roll_no, s0.name)

s1 = Student()
print(s1.roll_no, s1.name)

s2 = Student()
s2.roll_no = 2
s2.name = 'Harish'
print(s2.roll_no, s2.name)

s50 = Student()
s5
s50.name = 'Asmita'
print(s50.roll_no, s50.name)
```

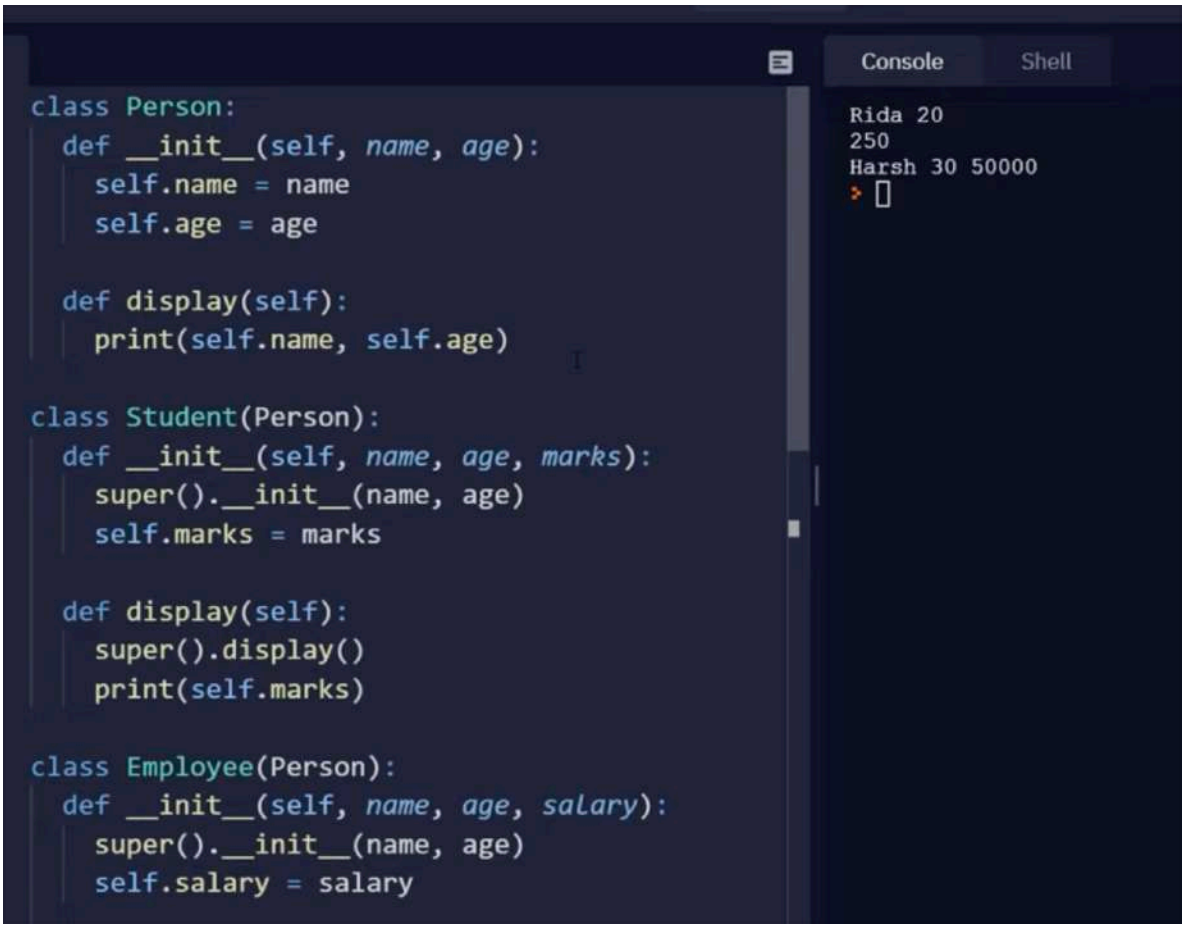
```
0 Bhuvanesh
None None
2 Harish
None Asmita
>
```

ain.py

```python
    def __init__(self, roll_no, name, total):
        self.roll_no = roll_no
        self.name = name
        self.total = total

    def display(self):
        print(self.roll_no, self.name, self.total)

    def result(self):
        if self.total > 120:
            print('Pass')
        else:
            print('Fail')

s0 = Student(0, 'Bhuvanesh', 100)
s0.display()
s0.result()

s1 = Student(1, 'Harish', 150)
s1.display()
s1.result()
```

Console Shell

```
0 Bhuvanesh 100
Fail
1 Harish 150
Pass
>
```



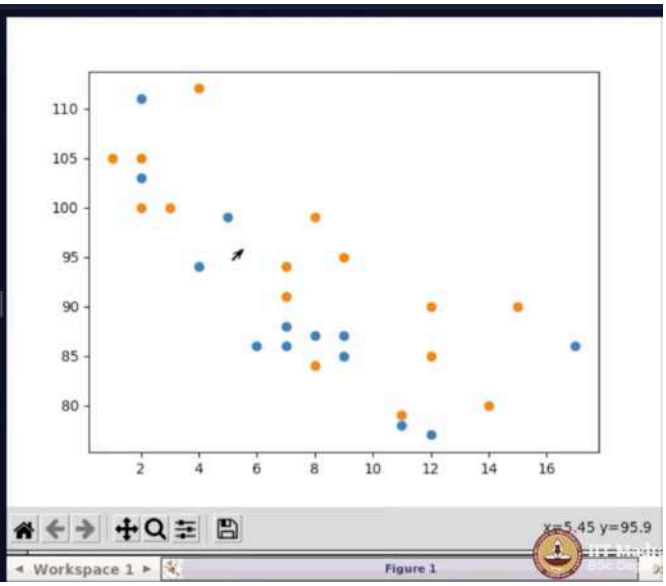
Parameters	Python list	NumPy array
Installation and importing	Not required	Required
Type of elements	Heterogenous	Homogenous
Dimension of elements	No restriction	Has to be same
Memory allocation	Non-contiguous	Contiguous
Size	Requires more space	Requires less space
Performance	Slower	Faster
Element wise operations	Not possible	Possible
Functionality	Can not handle arithmetic operations	Can handle arithmetic operations

```python
import numpy as np

a = np.array(42)
b = np.array([1, 2, 3, 4, 5])
c = np.array([[1, 2, 3], [4, 5, 6]])
d = np.array([[[1, 2, 3], [4, 5, 6]], [[1, 2, 3], [4, 5, 6]]])

print(a, a.ndim, '\n')
print(b, b.ndim, '\n')
print(c, c.ndim, '\n')
print(d, d.ndim, '\n')
```

```
Console  Shell
42 0

[1 2 3 4 5] 1

[[1 2 3]
 [4 5 6]] 2

[[[1 2 3]
  [4 5 6]]

 [[1 2 3]
  [4 5 6]]] 3
```

main.py

```python
import matplotlib.pyplot as plt
import numpy as np

x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y)

x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y)

plt.show()
```



main.py

```python
a = int(input())
b = int(input())
try:
    f = open('abc.txt','r')
    c = a / b
    print(c)
except ZeroDivisionError:
    print('Invalid input, divisor can not be zero')
except NameError:
    print('Variable not defined')
except FileNotFoundError:
    print('Invalid file name. Please check again')
except:
    print('Something went wrong')
```

```
Console  Shell
5
7
0.7142857142857143
Invalid file name. Please check again
```

```python
a = int(input())
b = int(input())
try:
    f = open('abc.txt','r')
    c = a / b
    print(c)
except ZeroDivisionError:
    print('Invalid input, divisor can not be zero')
except:
    print('Something went wrong')
finally:
    f.close()
    print('From finally block')
```

```
30
0
Invalid input, divisor can not be zero
From finally block
>
```

```python
a = int(input())
if a < 18:
    raise Exception('You are underage, can not vote')
```

```
3
Traceback (most recent call last):
  File "main.py", line 3, in <module>
    raise Exception('You are underage, can not vote')
Exception: You are underage, can not vote
>
```

```python
fruits = ["mango", "apple", "banana",
"orange", "pineapple", "watermelon", "guava",
"kiwi"]

bascket = iter(fruits)

print(next(bascket))



print(next(bascket))


print(next(bascket))


print(next(bascket))


print(next(bascket))
```

Console Shell

```
mango
apple
banana
orange
pineapple
>
```

RE VIDEOS

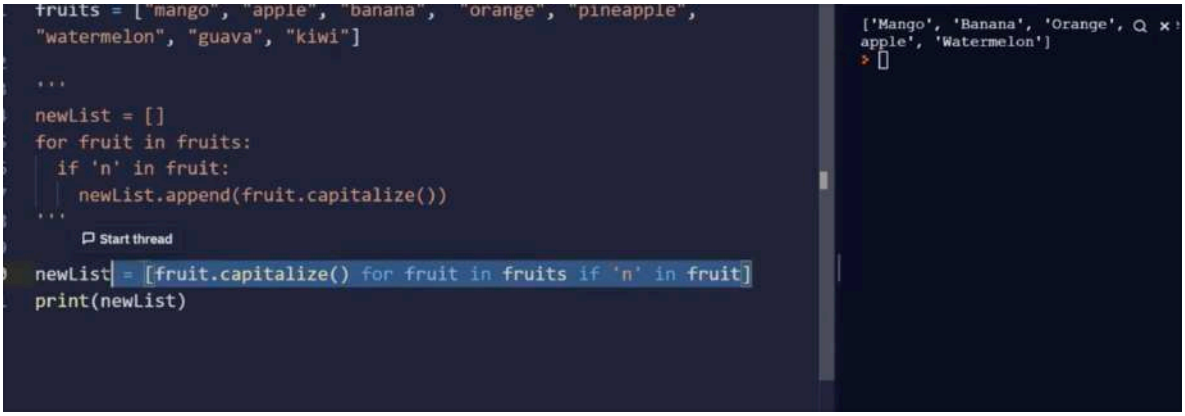
IIT BSc

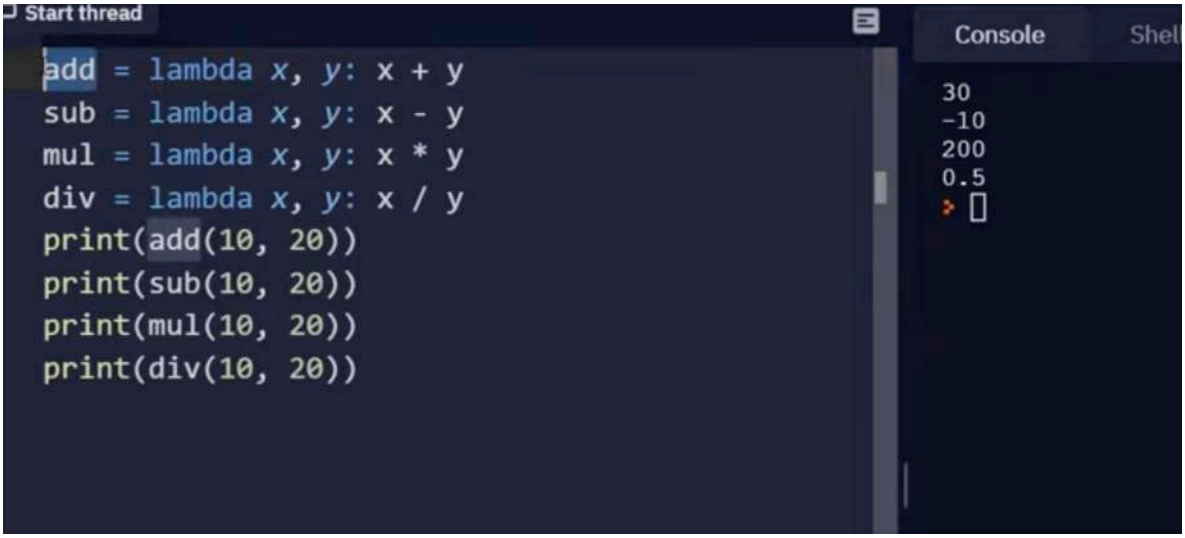
```python
def square(limit):
    x = 0
    while x < limit:
        yield x * x
        yield x * x * x
        x += 1

a = square(5)
print(next(a), next(a))
print(next(a), next(a))
print(next(a), next(a))
print(next(a), next(a))
```

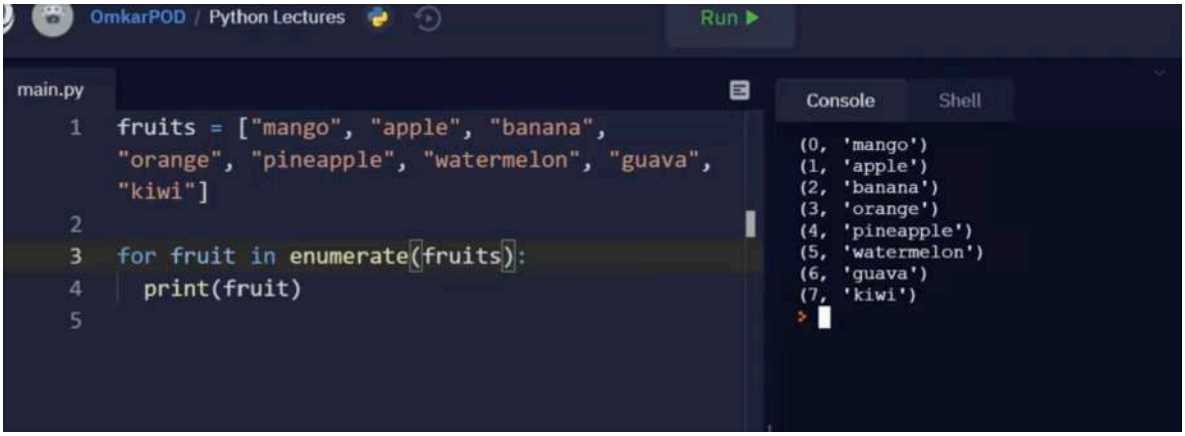
Start thread

Console Shell

```
0 0
1 1
4 8
9 27
>
```


fruits = ["mango", "apple", "banana", "orange", "pineapple",
"watermelon", "guava", "kiwi"]
'''
newList = []
for fruit in fruits:
if 'n' in fruit:
newList.append(fruit.capitalize())
'''
Start thread
newList = [fruit.capitalize() for fruit in fruits if 'n' in fruit]
print(newList)
['Mango', 'Banana', 'Orange',
apple', 'Watermelon']


Start thread
add = lambda x, y: x + y
sub = lambda x, y: x - y
mul = lambda x, y: x * y
div = lambda x, y: x / y
print(add(10, 20))
print(sub(10, 20))
print(mul(10, 20))
print(div(10, 20))
Console
Shell
30
-10
200
0.5


OmkarPOD / Python Lectures
Run
main.py
fruits = ["mango", "apple", "banana",
"orange", "pineapple", "watermelon", "guava",
"kiwi"]
for fruit in enumerate(fruits):
print(fruit)
Console
Shell
(0, 'mango')
(1, 'apple')
(2, 'banana')
(3, 'orange')
(4, 'pineapple')
(5, 'watermelon')
(6, 'guava')
(7, 'kiwi')


fruits = ["mango", "apple", "banana",
"orange", "pineapple", "watermelon", "guava",
"kiwi"]
size = [5, 5, 6, 6, 9, 10, 5, 4]
print(list(zip(fruits, size)))
Console
Shell
[('mango', 5), ('apple', 5), ('banana', 6), ('orange', 6),
apple', 9), ('watermelon', 10), ('guava', 5), ('kiwi', 4)]

```python
a = [10, 20, 30, 40, 50, 60]
b = [5, 10, 15, 20, 25, 30]
#c = a + 1
def sub(x, y):
  return x - y
def incr(x):
  return x + 1
c = map(sub, a, b)
c = map(incr, a)
print(list(c))
```

```
[11, 21, 31, 41, 51, 61]
>
```

```python
import math

a = [25, -16, 9, 81, -100]

def square_root(n):
  return math.sqrt(n)

def is_positive(n):
  if n >= 0:
    return n

c = map(square_root, filter(is_positive, a))
print(list(c))
```

```
[5.0, 3.0, 9.0]
>
```