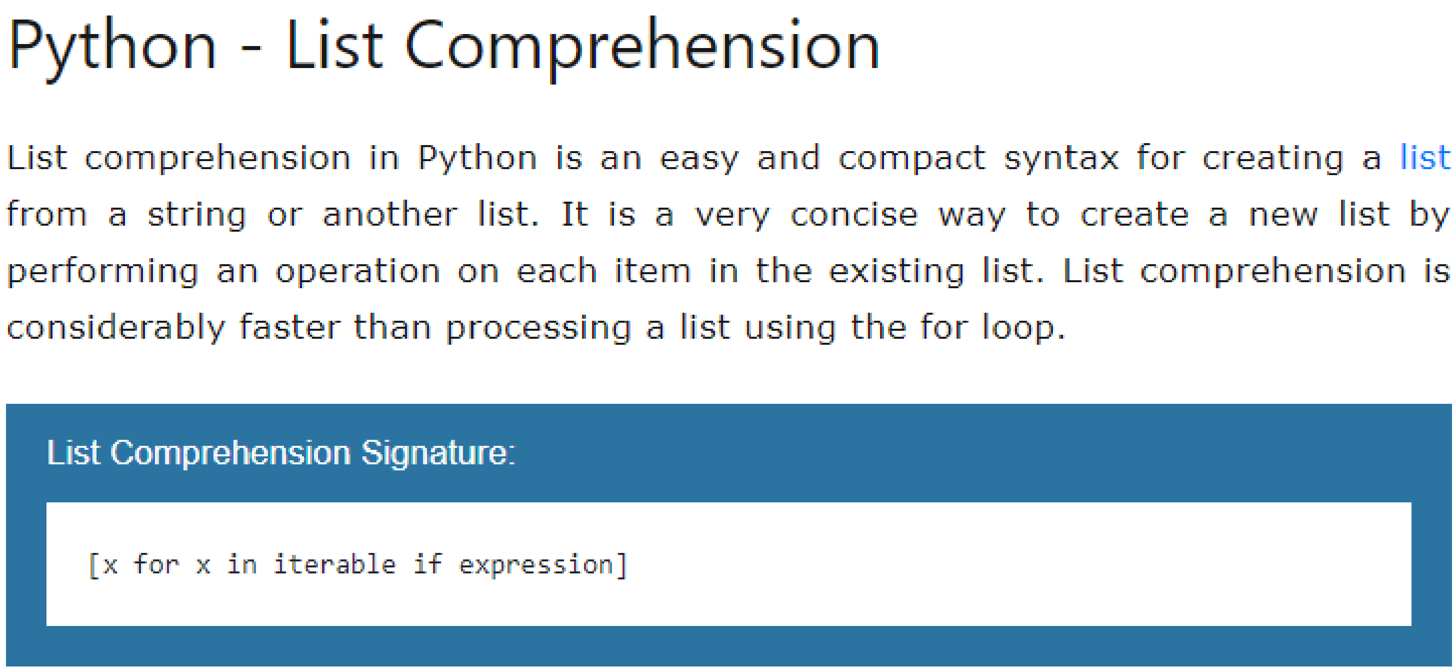
Chapter 8:

Data Structures -1 Linear Lists (E)

By: Jagdish Devrani
Lecturer in Computer Science
Ahlcon Public Shool

Index

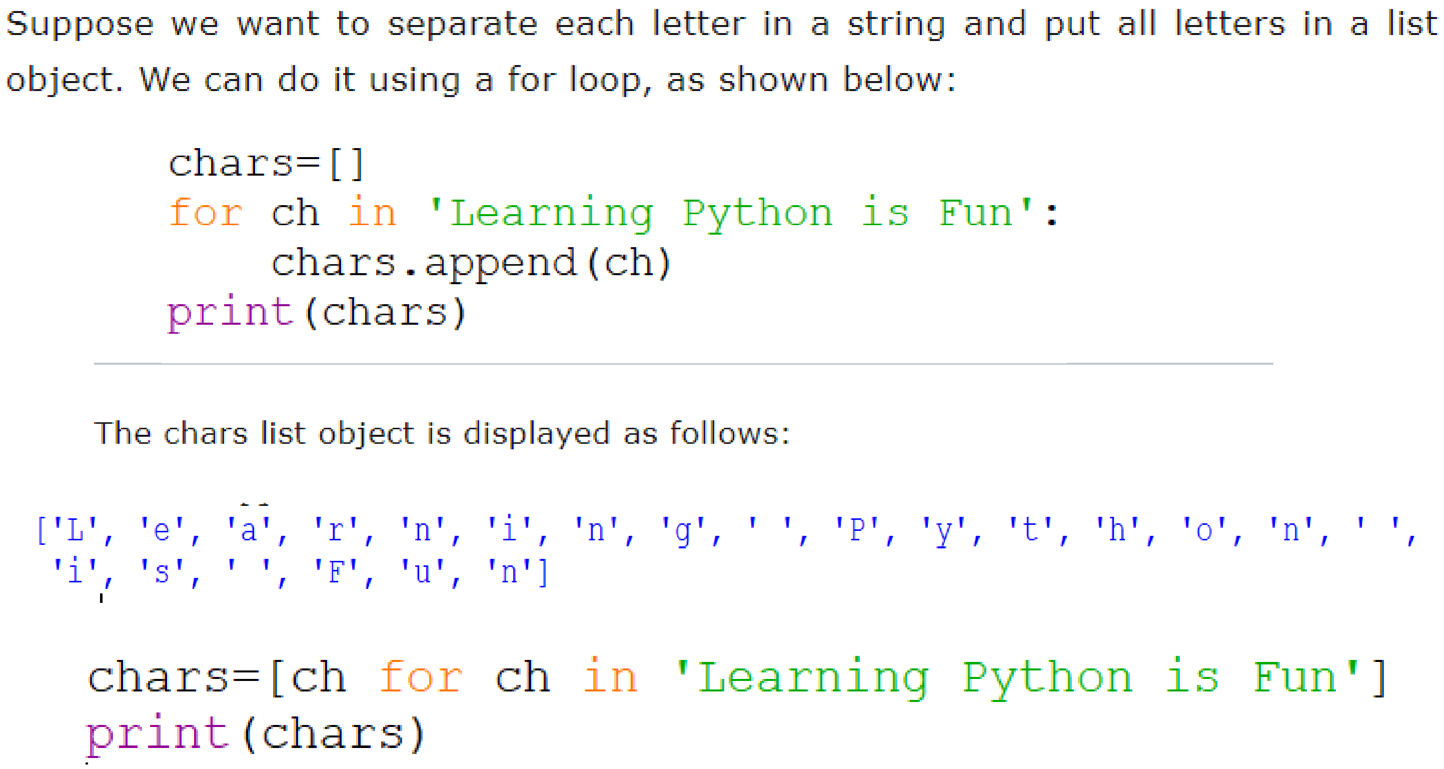


Python - List Comprehension

List comprehension in Python is an easy and compact syntax for creating a list from a string or another list. It is a very concise way to create a new list by performing an operation on each item in the existing list. List comprehension is considerably faster than processing a list using the for loop.

List Comprehension Signature:

```
[x for x in iterable if expression]
```



Suppose we want to separate each letter in a string and put all letters in a list object. We can do it using a for loop, as shown below:

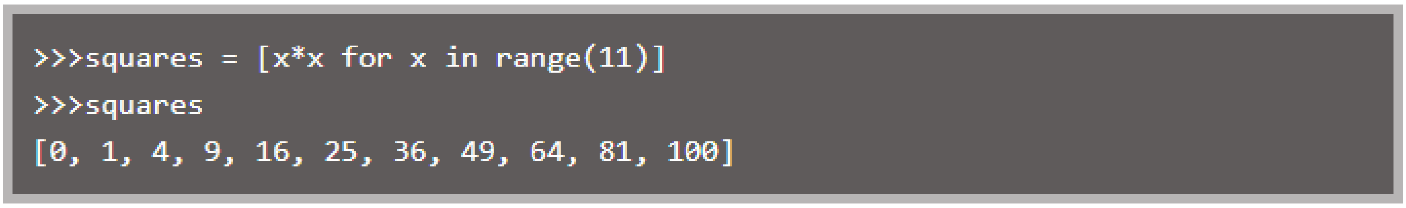
```
chars=[]
for ch in 'Learning Python is Fun':
    chars.append(ch)
print(chars)
```

The chars list object is displayed as follows:

```
['L', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ' ', 'P', 'y', 't', 'h', 'o', 'n', ' ',
 'i', 's', ' ', 'F', 'u', 'n']
```

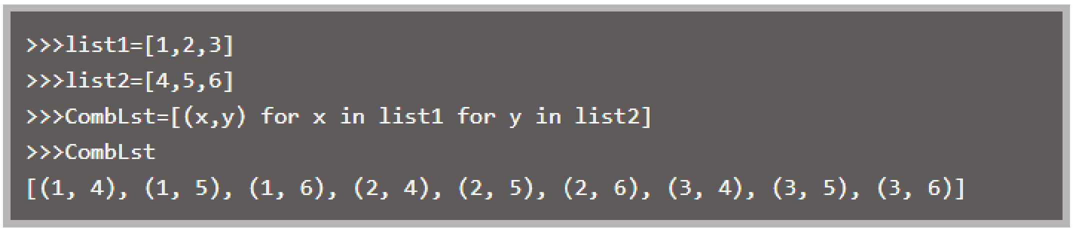
```
chars=[ch for ch in 'Learning Python is Fun']
print(chars)
```

The following example uses a list comprehension to build a list of squares of the numbers between 1 and 10.



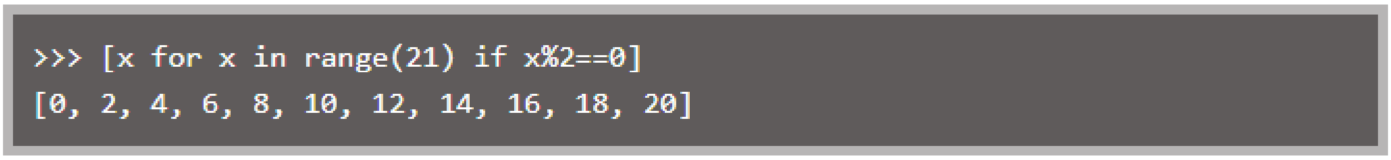
```
>>>squares = [x*x for x in range(11)]
>>>squares
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

It is possible to use nested loops in a list comprehension expression. In the following example, all combinations of items from two lists in the form of a tuple are added in a third list object.



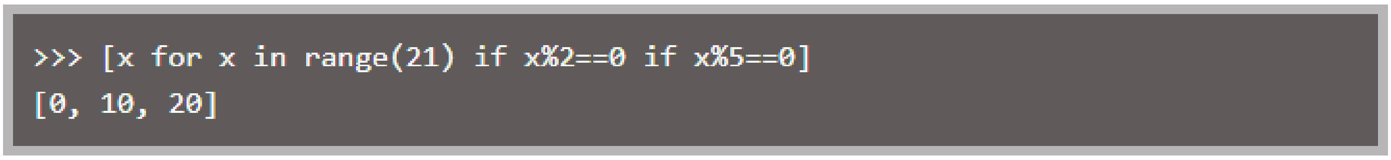
```
>>>list1=[1,2,3]
>>>list2=[4,5,6]
>>>CombLst=[(x,y) for x in list1 for y in list2]
>>>CombLst
[(1, 4), (1, 5), (1, 6), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6)]
```

We can even have the if condition in a list comprehension. The following statement will result in a list of all even numbers between 1 and 20.

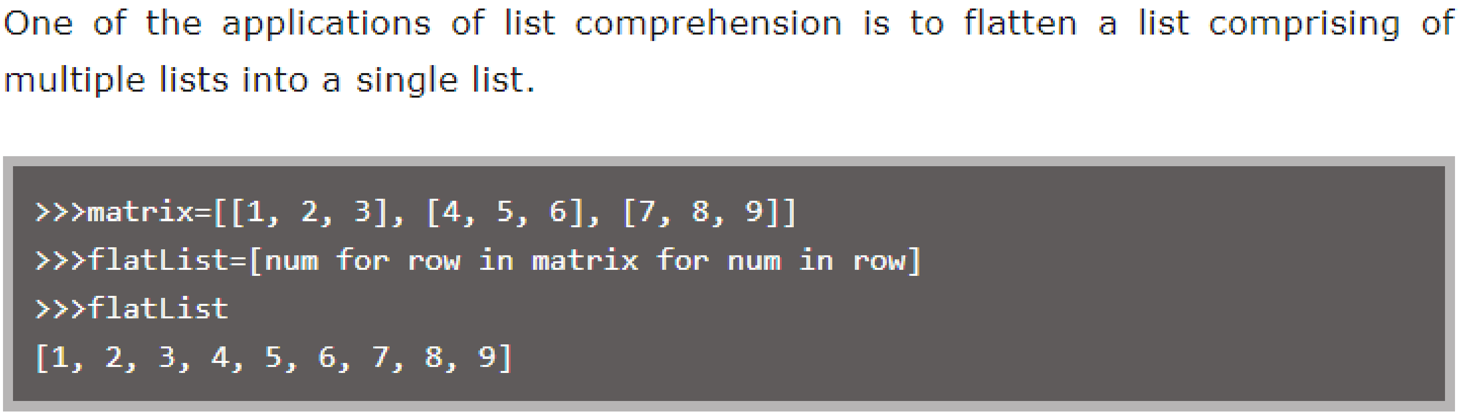


```
>>> [x for x in range(21) if x%2==0]
[0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

We can use nested if conditions with list comprehension.



```
>>> [x for x in range(21) if x%2==0 if x%5==0]
[0, 10, 20]
```

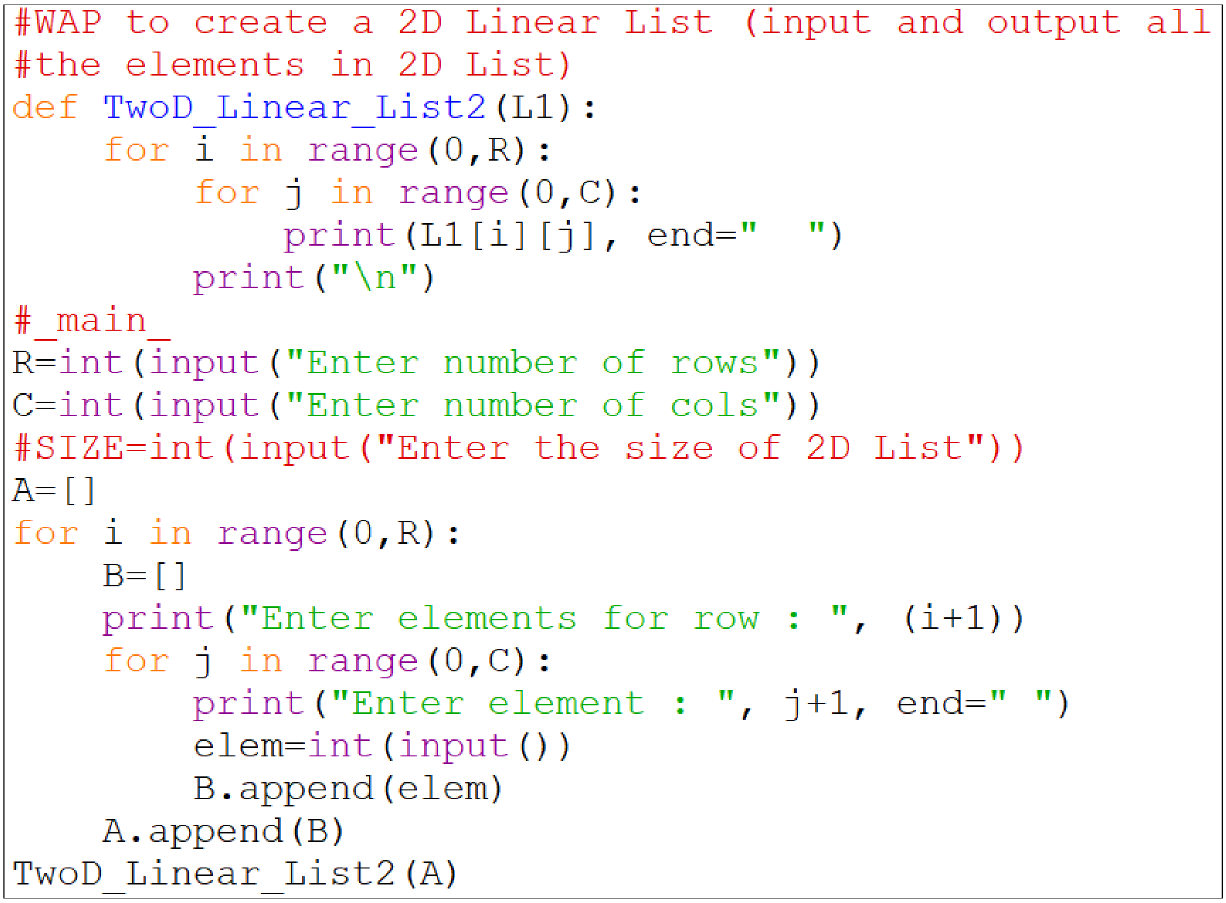


One of the applications of list comprehension is to flatten a list comprising of multiple lists into a single list.

```
>>>matrix=[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
>>>flatList=[num for row in matrix for num in row]
>>>flatList
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Advantage of List Comprehensions

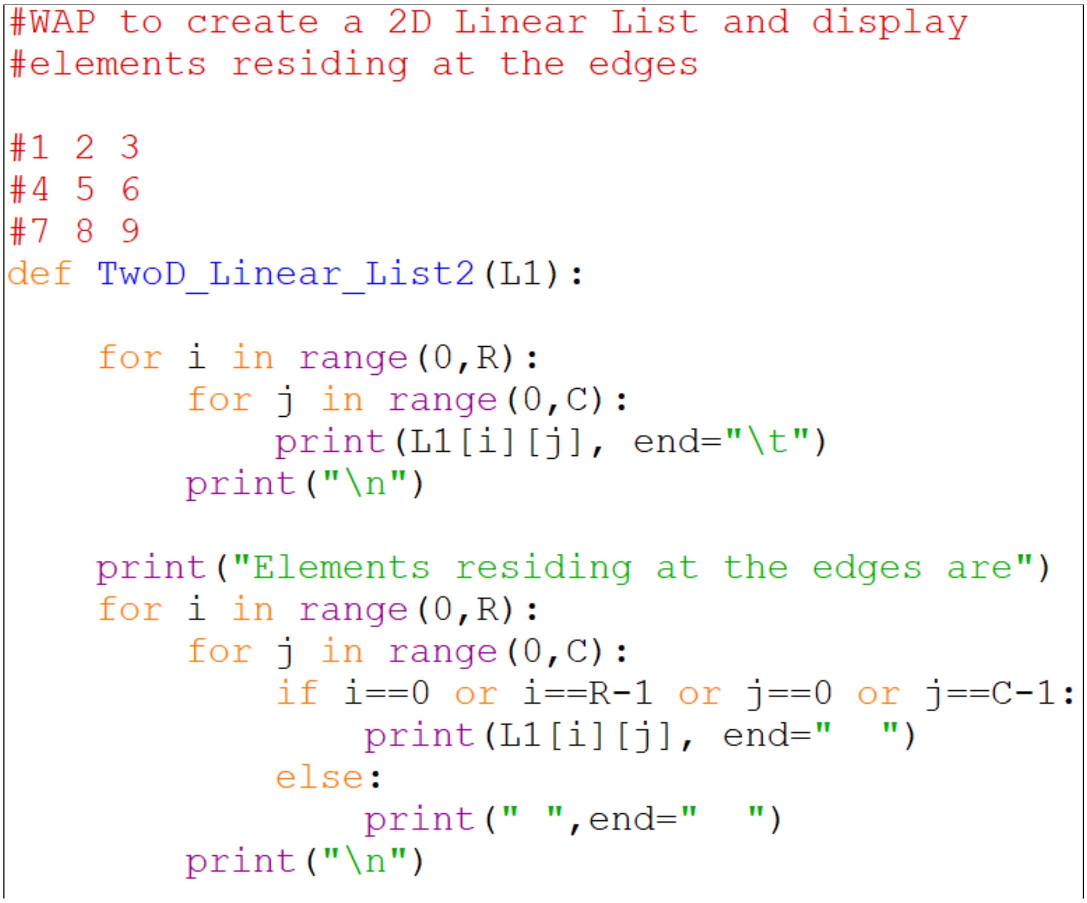
1) Code Reduction
2) Faster code processing



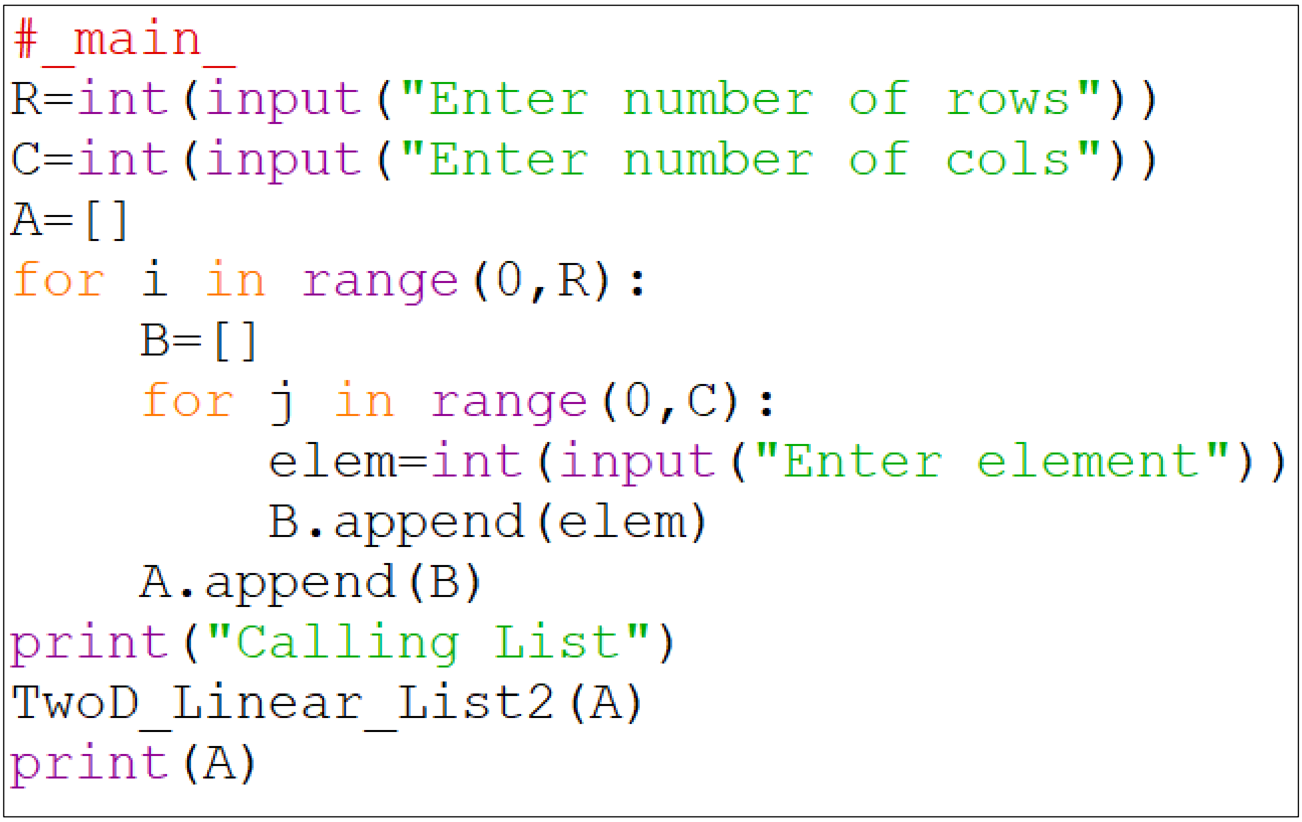
```
#WAP to create a 2D Linear List (input and output all
#the elements in 2D List)
def TwoD_Linear_List2(L1):
    for i in range(0,R):
        for j in range(0,C):
            print(L1[i][j], end="   ")
        print("\n")
#_main_
R=int(input("Enter number of rows"))
C=int(input("Enter number of cols"))
#SIZE=int(input("Enter the size of 2D List"))
A=[]
for i in range(0,R):
    B=[]
    print("Enter elements for row : ", (i+1))
    for j in range(0,C):
        print("Enter element : ", j+1, end=" ")
        elem=int(input())
        B.append(elem)
    A.append(B)
TwoD_Linear_List2(A)
```

```
#WAP to create a 2D Linear List
#(input and output all the elements in 2D List)
def TwoD_Linear_List2(L1):
    for i in range(0,R):
        for j in range(0,C):
            print(L1[i][j], end="\t")
        print("\n")
    if R==C:
        print("Diagonal Elements are")
        for i in range(0,R):
            for j in range(0,C):
                if i==j or i+j==R-1:
                    print(L1[i][j], end="  ")
                else:
                    print("",end="  ")
            print("\n")
    else:
        print("Diagonal Elements can not be displayed")
```

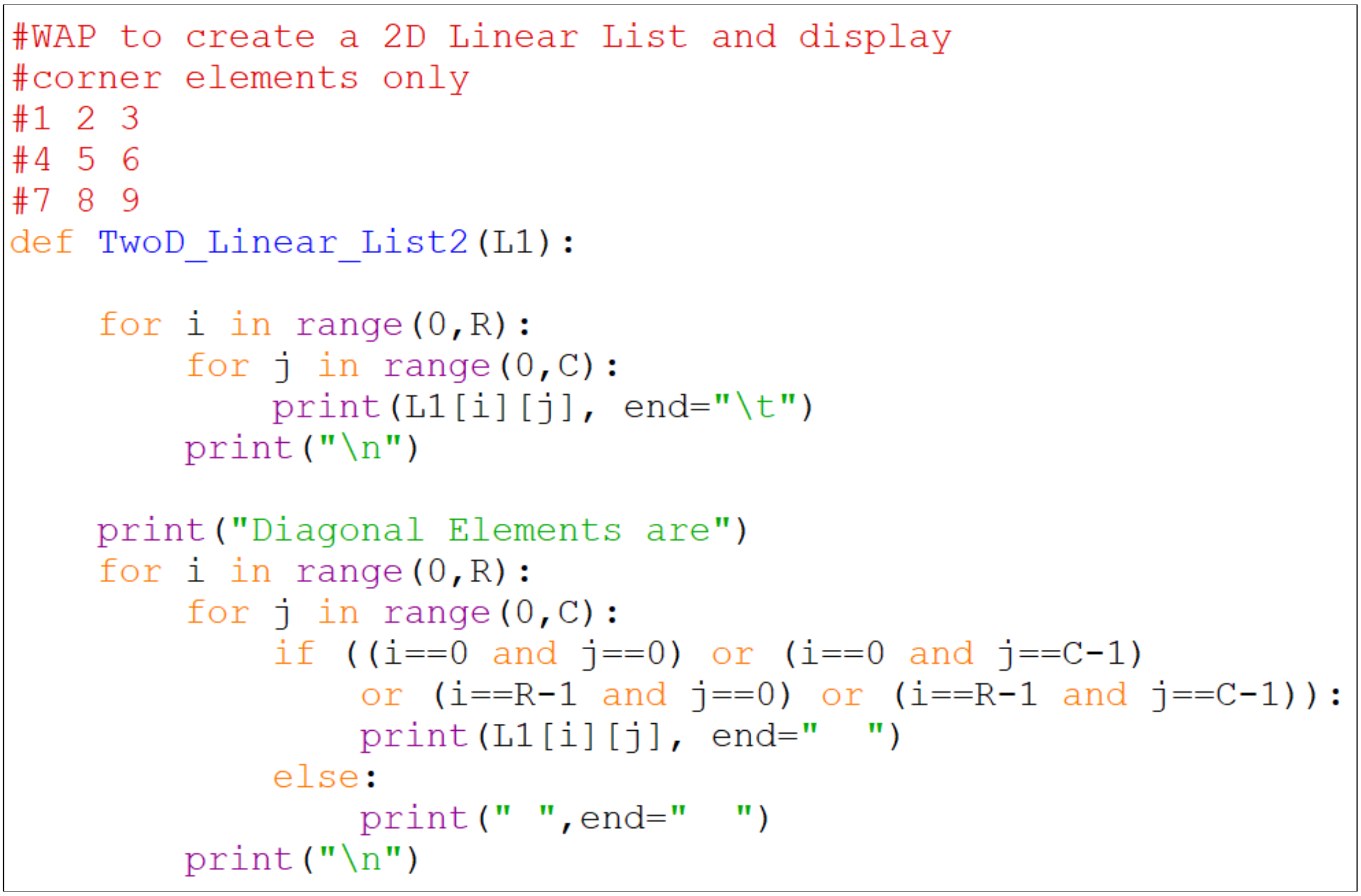
```
#_main_
R=int(input("Enter number of rows"))
C=int(input("Enter number of cols"))
if R==C:
    A=[]
    for i in range(0,R):
        B=[]
        for j in range(0,C):
            elem=int(input("Enter element"))
            B.append(elem)
        A.append(B)
    print("Calling List")
    TwoD_Linear_List2(A)
    print(A)
else:
    print("Diagonal Elements can not be displayed")
```



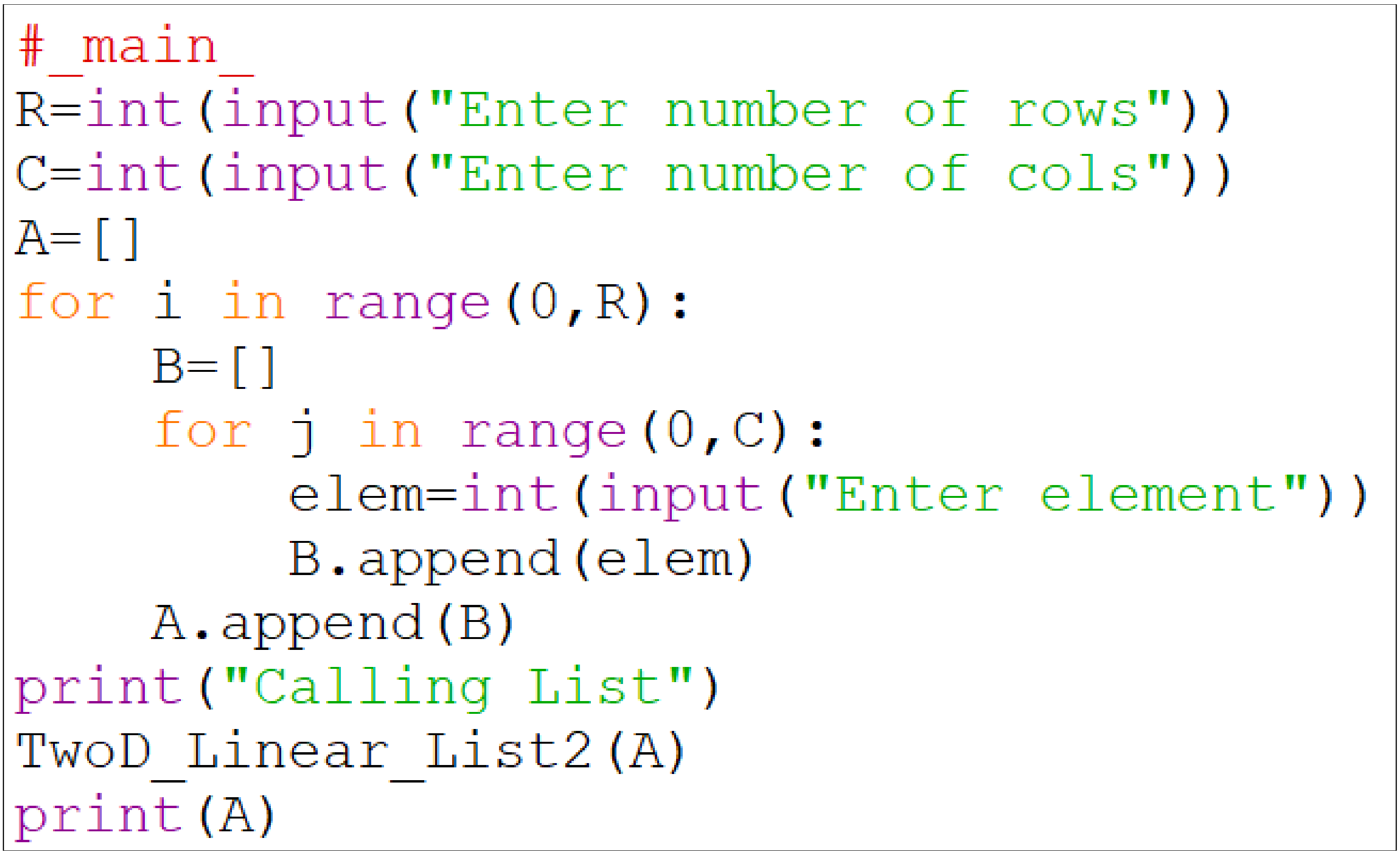
```
#WAP to create a 2D Linear List and display
#elements residing at the edges

#1  2  3
#4  5  6
#7  8  9
def TwoD_Linear_List2(L1):

    for i in range(0,R):
        for j in range(0,C):
            print(L1[i][j], end="\t")
        print("\n")

    print("Elements residing at the edges are")
    for i in range(0,R):
        for j in range(0,C):
            if i==0 or i==R-1 or j==0 or j==C-1:
                print(L1[i][j], end="   ")
            else:
                print(" ",end="   ")
        print("\n")
```



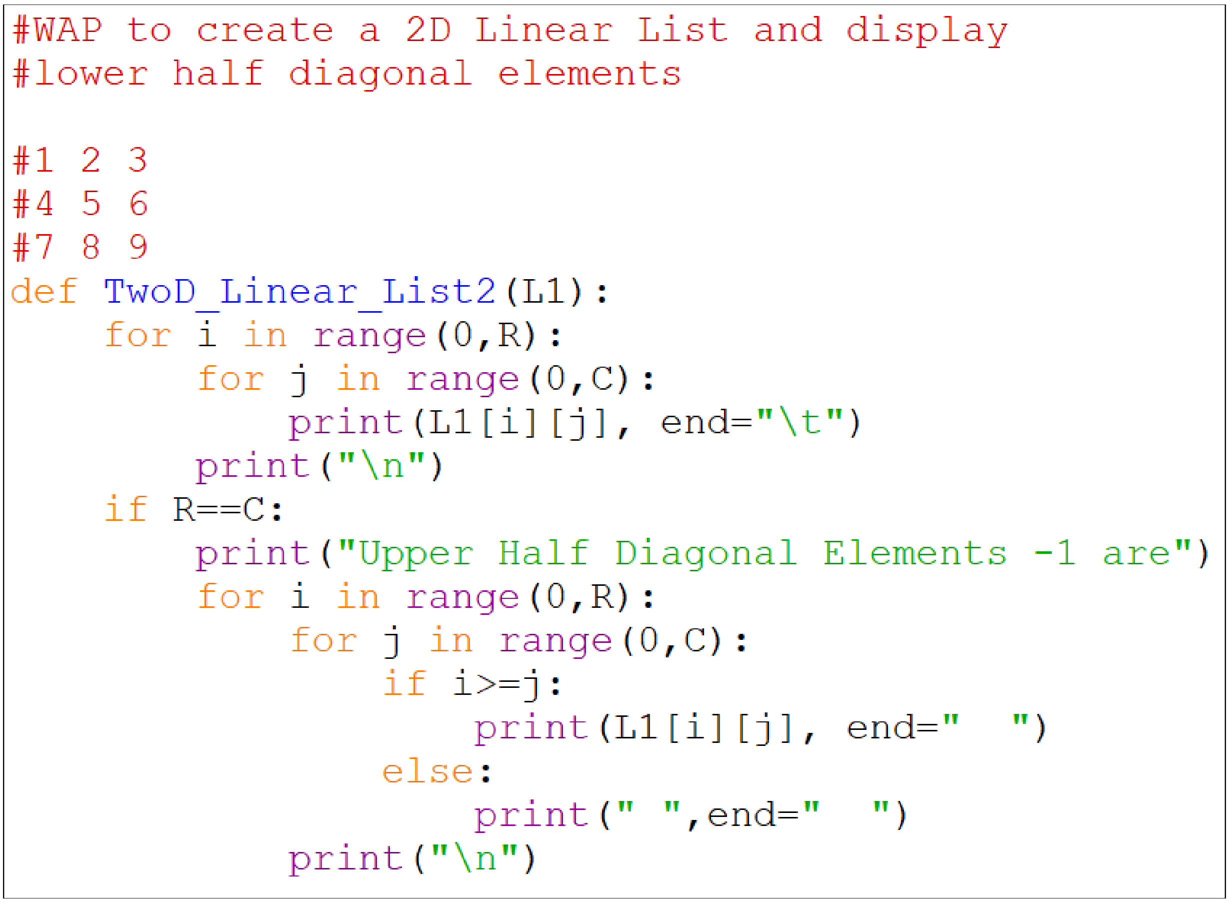
```
#_main_
R=int(input("Enter number of rows"))
C=int(input("Enter number of cols"))
A=[]
for i in range(0,R):
    B=[]
    for j in range(0,C):
        elem=int(input("Enter element"))
        B.append(elem)
    A.append(B)
print("Calling List")
TwoD_Linear_List2(A)
print(A)
```



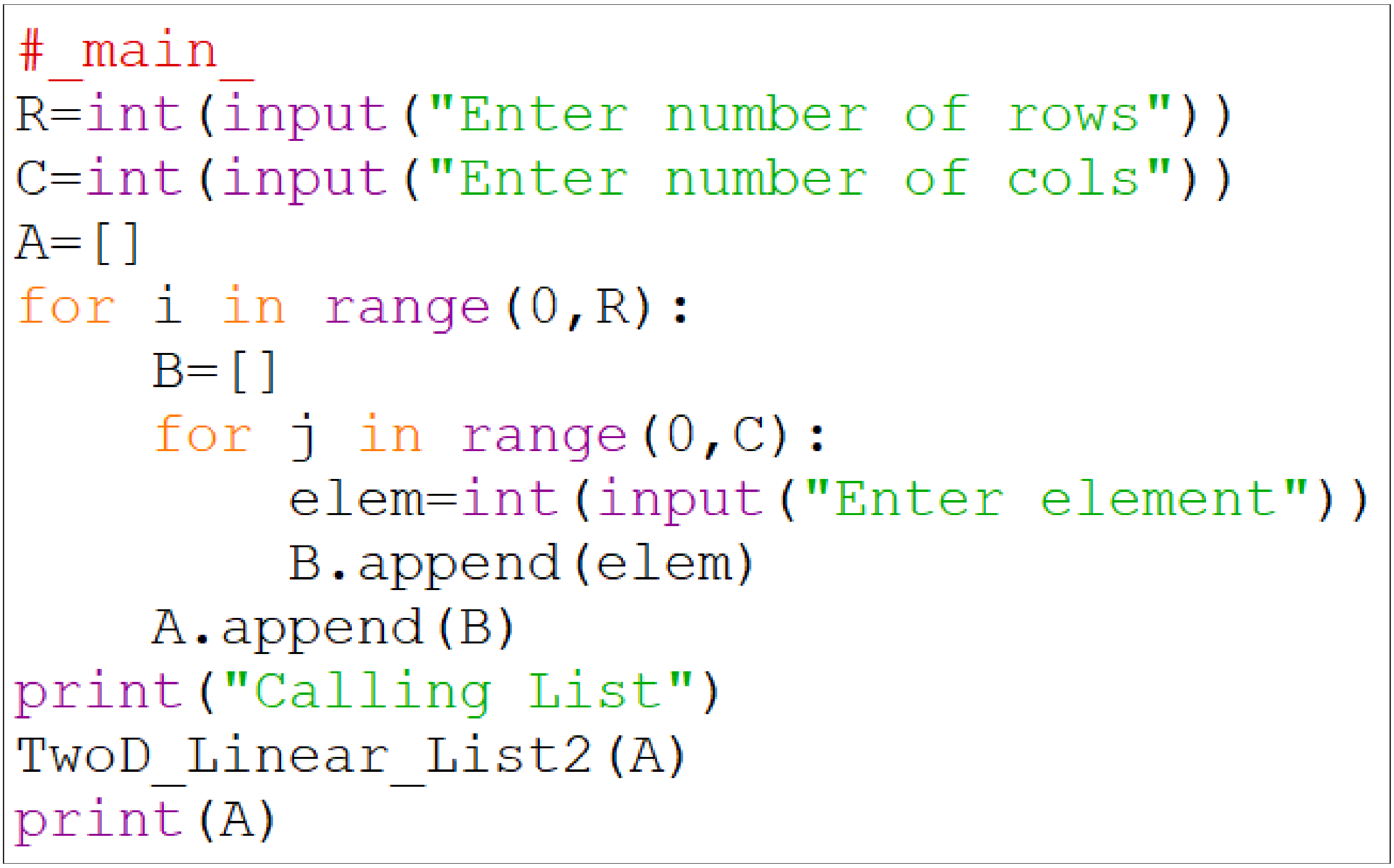
```
#WAP to create a 2D Linear List and display
#corner elements only
#1 2 3
#4 5 6
#7 8 9
def TwoD_Linear_List2(L1):

    for i in range(0,R):
        for j in range(0,C):
            print(L1[i][j], end="\t")
        print("\n")

    print("Diagonal Elements are")
    for i in range(0,R):
        for j in range(0,C):
            if ((i==0 and j==0) or (i==0 and j==C-1)
                or (i==R-1 and j==0) or (i==R-1 and j==C-1)):
                print(L1[i][j], end="  ")
            else:
                print("  ",end="  ")
        print("\n")
```



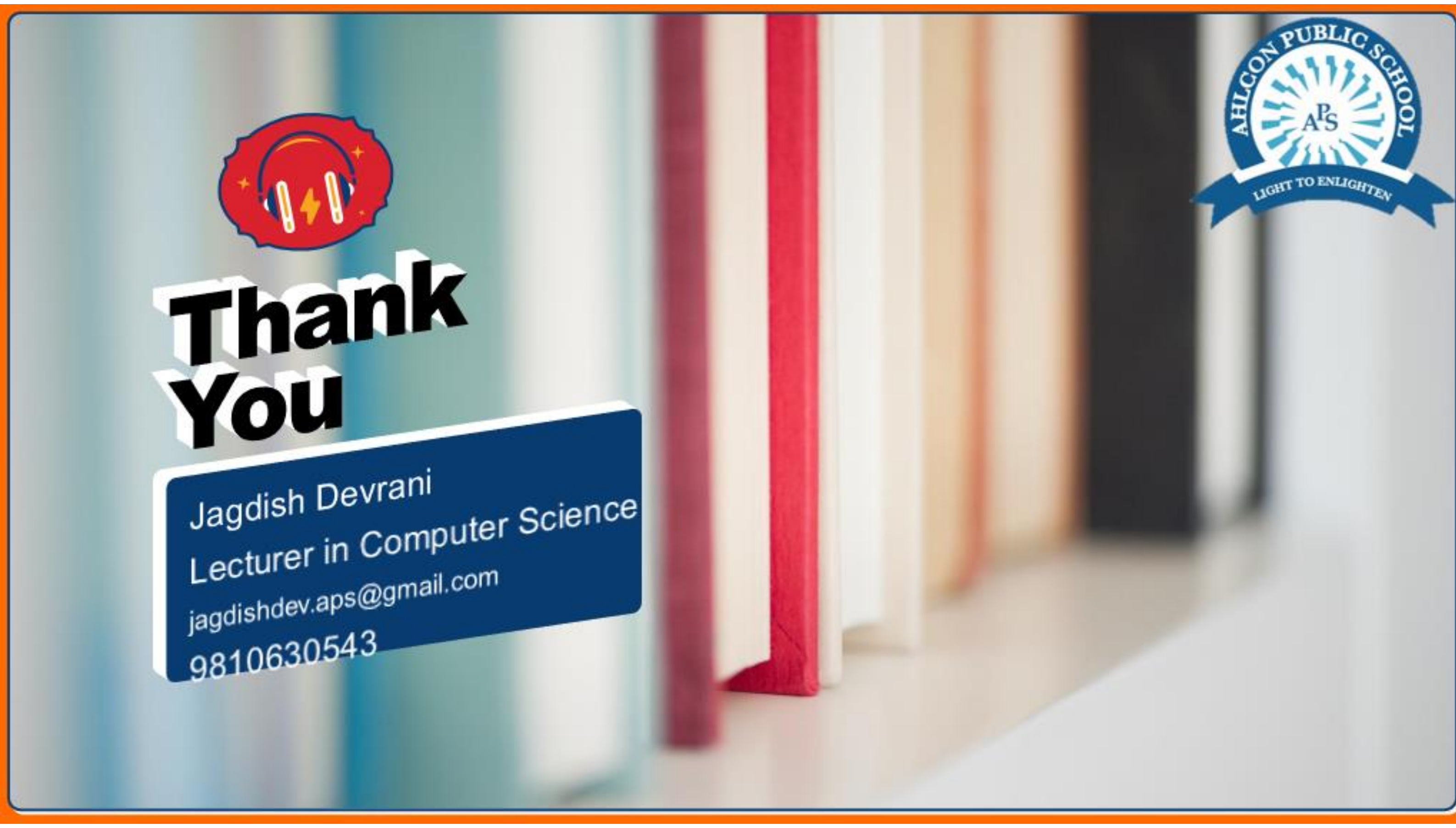
```
#_main_
R=int(input("Enter number of rows"))
C=int(input("Enter number of cols"))
A=[]
for i in range(0,R):
    B=[]
    for j in range(0,C):
        elem=int(input("Enter element"))
        B.append(elem)
    A.append(B)
print("Calling List")
TwoD_Linear_List2(A)
print(A)
```



```
#WAP to create a 2D Linear List and display
#lower half diagonal elements

#1  2  3
#4  5  6
#7  8  9
def TwoD_Linear_List2(L1):
    for i in range(0,R):
        for j in range(0,C):
            print(L1[i][j], end="\t")
        print("\n")
    if R==C:
        print("Upper Half Diagonal Elements -1 are")
        for i in range(0,R):
            for j in range(0,C):
                if i>=j:
                    print(L1[i][j], end="  ")
                else:
                    print(" ",end="  ")
            print("\n")
```



```
#_main_
R=int(input("Enter number of rows"))
C=int(input("Enter number of cols"))
A=[]
for i in range(0,R):
    B=[]
    for j in range(0,C):
        elem=int(input("Enter element"))
        B.append(elem)
    A.append(B)
print("Calling List")
TwoD_Linear_List2(A)
print(A)
```



AHLCON PUBLIC SCHOOL

LIGHT TO ENLIGHTEN

Thank You

Jagdish Devrani
Lecturer in Computer Science
jagdishdev.aps@gmail.com
9810630543