

PRACTICAL: 3

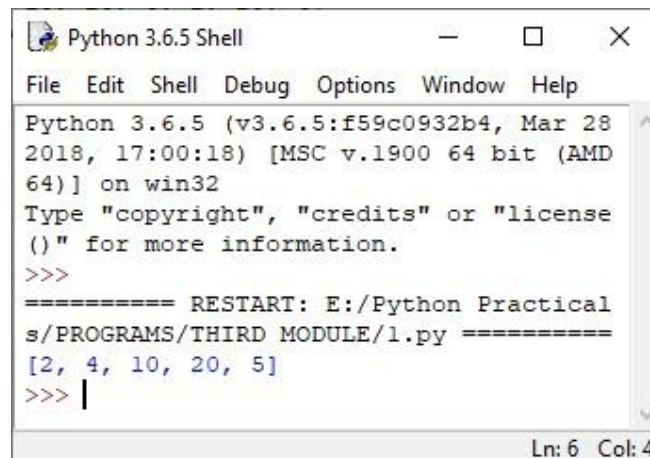
Develop programs to learn list and tuple in python.

PROGRAM 1: Write a program to remove duplicates from list.

SOLUTION:

```
def Remove(duplicate):
    final_list = []
    for num in duplicate:
        if num not in final_list:
            final_list.append(num)
    return final_list
duplicate = [2, 4, 10, 20, 5, 2, 20, 4]
print(Remove(duplicate))
```

OUTPUT:



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD 64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/Python Practical
s/PROGRAMS/THIRD MODULE/1.py =====
[2, 4, 10, 20, 5]
>>> |
```

PROGRAM 2: Write a program to find frequency of elements of list.

SOLUTION:

```
def CountFrequency(my_list):

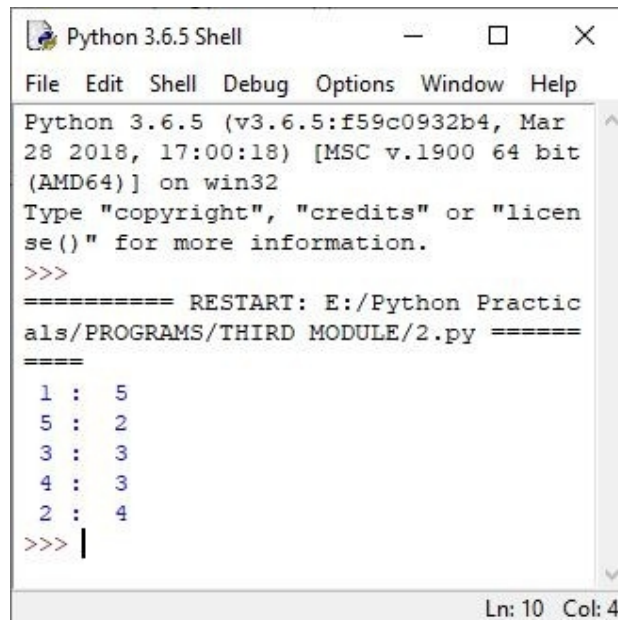
    # Creating an empty dictionary
    freq = {}
    for item in my_list:
        if (item in freq):
            freq[item] += 1
        else:
            freq[item] = 1

    for key, value in freq.items():
        print ("% d : % d"%(key, value))

if __name__ == "__main__":
    my_list =[1, 1, 1, 5, 5, 3, 1, 3, 3, 1, 4, 4, 4, 2, 2, 2, 2]

    CountFrequency(my_list)
```

OUTPUT:



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/Python Practic
als/PROGRAMS/THIRD MODULE/2.py =====
=====
1 : 5
5 : 2
3 : 3
4 : 3
2 : 4
>>> |
```

PROGRAM 3: Write a program to sort given list - [23, 45, 36, 27, 98, 12, 56, 78, 85] without using built in function.

SOLUTION:

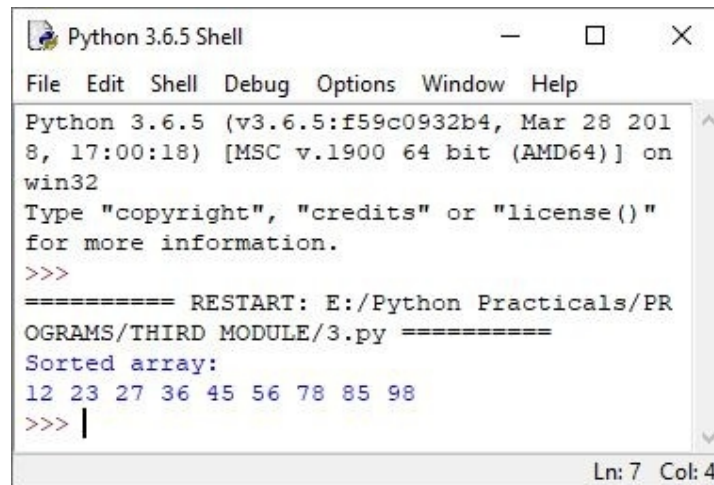
```
def sortArr(arr, n, min_no, max_no):
    # Count of elements in given range
    m = max_no - min_no + 1

    # Count frequencies of all elements
    c = [0] * m
    for i in range(n):
        c[arr[i] - min_no] += 1

    # Traverse through range. For every
    # element, print it its count times.
    print('Sorted array: ')
    for i in range(m):
        for j in range(c[i]):
            print((i + min_no), end=" ")

arr = [23, 45, 36, 27, 98, 12, 56, 78, 85]
min_no, max_no = 0, 100
n = len(arr)
sortArr(arr, n, min_no, max_no)
```

OUTPUT:



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/Python Practicals/PROGRAMS/THIRD MODULE/3.py =====
Sorted array:
12 23 27 36 45 56 78 85 98
>>> |
```

PROGRAM 4: Write a program for matrix multiplication and matrix addition using list.

SOLUTION:

1. For Matrix Addition:

```
X = [[1,2,3],
      [4,5,6],
      [7,8,9]]
```

```
Y = [[9,8,7],
      [6,5,4],
      [3,2,1]]
```

```
result = [[0,0,0],
           [0,0,0],
           [0,0,0]]
```

iterate through rows

for i in range(len(X)):

iterate through columns

for j in range(len(X[0])):

result[i][j] = X[i][j] + Y[i][j]

print('Matrix addition: ')

for r in result:

print(r)

2. For Matrix Multiplication:

```
X = [[1,2,3],
      [4,5,6],
      [7,8,9]]
```

```
Y = [[9,8,7],
      [6,5,4],
      [3,2,1]]
```

```
result = [[0,0,0],
          [0,0,0],
          [0,0,0]]

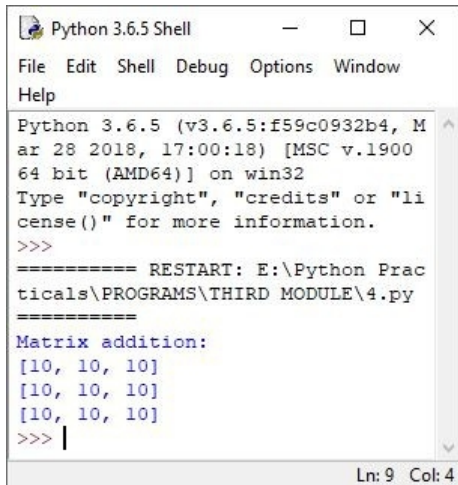
# iterating by row of X
for i in range(len(X)):

    # iterating by coloum by Y
    for j in range(len(Y[0])):

        # iterating by rows of Y
        for k in range(len(Y)):
            result[i][j] += X[i][k] * Y[k][j]
print('\nMatrix multiplication: ')
for r in result:
    print(r)
```

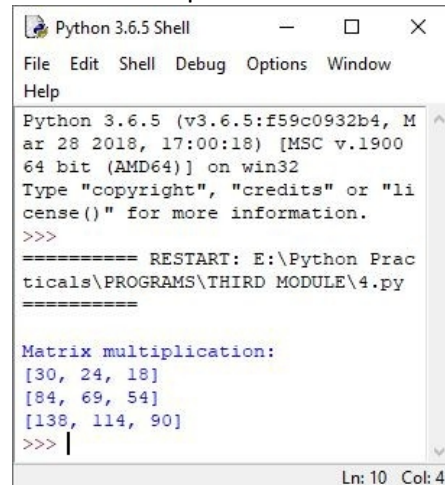
OUTPUT:

1. For Matrix Addition:



```
Python 3.6.5 (v3.6.5:f59c0932b4, M
ar 28 2018, 17:00:18) [MSC v.1900
64 bit (AMD64)] on win32
Type "copyright", "credits" or "li
cense()" for more information.
>>>
===== RESTART: E:\Python Prac
ticals\PROGRAMS\THIRD MODULE\4.py
=====
Matrix addition:
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
>>> |
```

2. For Matrix Multiplication:



```
Python 3.6.5 (v3.6.5:f59c0932b4, M
ar 28 2018, 17:00:18) [MSC v.1900
64 bit (AMD64)] on win32
Type "copyright", "credits" or "li
cense()" for more information.
>>>
===== RESTART: E:\Python Prac
ticals\PROGRAMS\THIRD MODULE\4.py
=====
Matrix multiplication:
[30, 24, 18]
[84, 69, 54]
[138, 114, 90]
>>> |
```

PROGRAM 5: Write a program for generating Pascal's triangle using list.

SOLUTION:

```
from math import factorial
# input n
n = 5
for i in range(n):
    for j in range(n-i+1):

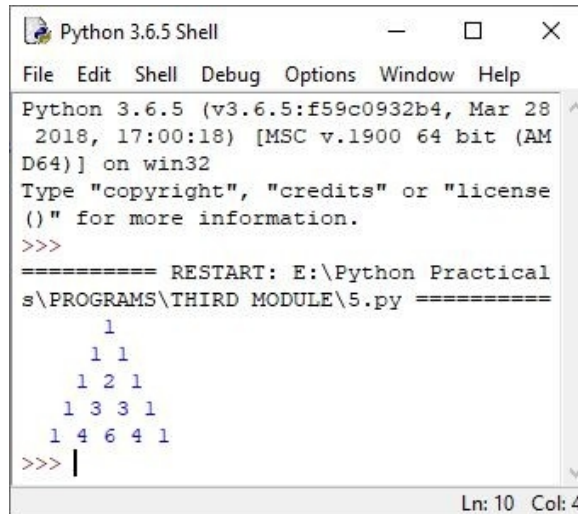
        # for left spacing
        print(end=" ")

    for j in range(i+1):

        # nCr = n!/((n-r)!*r!)
        print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")

    # for new line
    print()
```

OUTPUT:



```
Python 3.6.5 Shell
File Edit Shell Debug Options Window Help
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\Python Practical
s\PROGRAMS\THIRD MODULE\5.py =====
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
>>> |
```