In [1]:
```
#31.Create a list of tuples from given list having number and its cube in each tuple
list=[10,20,30,40]
res=[(val,pow(val,3)) for val in list]
print(res)
```

```
[(10, 1000), (20, 8000), (30, 27000), (40, 64000)]
```

In [48]:
```
#32.Python Sort Python Dictionaries by Key or value
my_dict={'c': 3,'a': 1,'b': 2}
sorted_dict={k: v for k, v in sorted(my_dict.items(), key=lambda item: item[0])}
print("sort dictionaries by key and value")
print(sorted_dict)
```

```
sort dictionaries by key and value
{'a': 1, 'b': 2, 'c': 3}
```

In [5]:
```
#33.Python dictionary with keys having multiple inputs
dic = {}
a,b,c= 5, 3, 10
p,q,r= 12, 6, 9
dic["x-y+z"] = [a-b+c,p-q+r]
print(dic)
```

```
{'x-y+z': [12, 15]}
```

In [6]:
```
#34.Python program to find the sum of all items in a dictionary
dic={ 'x':5, 'y':50, 'z':500, 'p':5000 }
print("Dictionary: ", dic)
#using sum() and values()
print("sum: ",sum(dic.values()))
```

```
Dictionary:  {'x': 5, 'y': 50, 'z': 500, 'p': 5000}
sum:  5555
```

In [7]:
```
#35.Python program to find the size of a Dictionary
import sys
dic1 = {"A": 1, "B": 2, "C": 3}
dic2 = {"Geek1": "python", "Geek2": "programming", "Geek3": "language"}
dic3 = {1: "Lion", 2: "Tiger", 3: "Fox", 4: "Wolf"}
print("Size of dic1: " + str(sys.getsizeof(dic1)) + "bytes")
print("Size of dic2: " + str(sys.getsizeof(dic2)) + "bytes")
print("Size of dic3: " + str(sys.getsizeof(dic3)) + "bytes")
```

```
Size of dic1: 232bytes
Size of dic2: 232bytes
Size of dic3: 232bytes
```

In [8]:
```
#36.Find the size of a Set in Python
import sys
Set1 = {"A", 1, "B", 2, "C", 3}
Set2 = {"Geek1", "Raju", "Geek2", "Nikhil", "Geek3", "Deepanshu"}
Set3 = {(1, "Lion"), ( 2, "Tiger"), (3, "Fox")}
print("Size of Set1: " + str(sys.getsizeof(Set1)) + "bytes")
print("Size of Set2: " + str(sys.getsizeof(Set2)) + "bytes")
print("Size of Set3: " + str(sys.getsizeof(Set3)) + "bytes")
```

```
Size of Set1: 472bytes
Size of Set2: 472bytes
Size of Set3: 216bytes
```

In [9]:
```
#37.Iterate over a set in Python
test_set = set("ViBgYoR")
for val in test_set:
 print(val)
```

```
V
B
g
Y
R
i
o
```

In [14]:
```
#38.Python – Maximum and Minimum in a Set
#maximun
def MAX(sets):
 return (max(sets))

sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])
print("the maximun element is:",MAX(sets))
def MIN(sets):
 return (min(sets))

sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])
print("the minimun element is:",MIN(sets))
```

```
the maximun element is: 65
the minimun element is: 1
```

In [15]:
```
#39.Python – Remove items from Set
colour={'pink','black','blue','white','red','green','orange'}
colour.remove('white')
print(colour)
```

```
{'orange', 'pink', 'blue', 'green', 'black', 'red'}
```

In [41]:
```
#40.Python – Check if two lists have atleast one element common
def common_data(list1,list2):
    result=False
    for x in list1:

        for y in list2:
            if x==y:
                result=True
                return result
    return result
a=[1,2,3,4,5]
b=[5,6,7,8,9]
print(common_data(a,b))
a=[1,2,3,4,5]
b=[6,7,8,9]
print(common_data(a,b))
```

```
True
False
```

In [26]:
```
#41.Python – Assigning Subsequent Rows to Matrix first row elements
test_list=[[5,8,9],[2,0,9],[5,4,2],[2,3,9]]
print("The original list:"+str(test_list))
res={test_list[0][ele]:test_list[ele+1] for ele in range(len(test_list)-1)}
print("the assigned matrix:"+str(res))
```

```
The original list:[[5, 8, 9], [2, 0, 9], [5, 4, 2], [2, 3, 9]]
the assigned matrix:{5: [2, 0, 9], 8: [5, 4, 2], 9: [2, 3, 9]}
```

In [27]:
```
#42.Adding and Subtracting Matrices in Python
# importing numpy as np
import numpy as np
#creating first matrix
A = np.array([[1, 2], [3, 4]])
# creating second matrix
B = np.array([[4, 5], [6, 7]])
print("Printing elements of first matrix")
print(A)
print("Printing elements of second matrix")
print(B)
# adding two matrix
print("Addition of two matrix")
print(np.add(A, B))
#subtracting two matrix
print("subtraction of two matrix")
print(np.subtract(a,b))
```

```
Printing elements of first matrix
[[1 2]
 [3 4]]
Printing elements of second matrix
[[4 5]
 [6 7]]
Addition of two matrix
[[ 5  7]
 [ 9 11]]
subtraction of two matrix
2
```

In [31]:
```
#43.Python – Group similar elements into Matrix
from itertools import groupby
test_list=[1,3,5,1,3,2,5,4,2]
print("The original list:" +str(test_list))
```

```
The original list:[1, 3, 5, 1, 3, 2, 5, 4, 2]
```

In [32]:
```
#44.Python – Row-wise element Addition in Tuple Matrix
test_list=[[('Gfg',3), ('is',3)], [('best',1)], [('for',5), ('geeks',1)]]
print("The original list is:"+ str(test_list))
cus_eles=[6,7,8]
res=[[sub+cus_eles[idx],) for sub in val] for idx, val in enumerate(test_list)]
print("The matrix after row elements addition :"+str(res))
```

```
The original list is:[[('Gfg', 3), ('is', 3)], [('best', 1)], [('for', 5), ('geeks', 1)]]
The matrix after row elements addition :[[('Gfg', 3, 6), ('is', 3, 6)], [('best', 1, 7)], [('for', 5, 8), ('geeks', 1, 8)]]
```

In [33]:
```
#45.Create an n x n square matrix, where all the sub-matrix has the sum of opposite corner elements as even
import itertools
def sub_mat_even(n):

    temp = itertools.count(1)


    l = [[next(temp)for i in range(n)]for i in range(n)]


    if n%2 == 0:
        for i in range(0,len(l)):
            if i%2 == 1:
                l[i][:] = l[i][::-1]


    for i in range(n):
        for j in range(n):
            print(l[i][j],end=" ")
        print()
n = 4
sub_mat_even(n)
```

```
1 2 3 4
8 7 6 5
9 10 11 12
16 15 14 13
```

In [34]:
```
#46.How to get list of parameters name from a function in Python
import inspect
import collections

print(inspect.signature(collections.Counter))
```

```
(iterable=None, /, **kwds)
```

In [35]:
```
#47.How to Print Multiple Arguments in Python
def GFG(name, num):
 print("Hello from ", name + ', ' + num)


GFG("geeks for geeks", "25")
```

```
Hello from  geeks for geeks, 25
```

In [40]:
```
#48.Python program to find the power of a number using recursion
def power(N, P):

    if P == 0:
        return 1

    return (N*power(N, P-1))

if __name__ == '__main__':
    N = 5
    P = 2
    print(power(N, P))
```

```
25
```

In [38]:
```
#49.Sorting objects of user defined class in Python
print(sorted([1,26,3,9]))

print(sorted("Geeks foR gEEks".split(), key=str.lower))
```

```
[1, 3, 9, 26]
['foR', 'Geeks', 'gEEks']
```

In [39]:
```
#50.Functions that accept variable length key value pair as arguments
def printValues(**kwargs):
    for key, value in kwargs.items():
        print("The value of {} is {}".format(key, value))
# driver code
if __name__ == '__main__':
    printValues(abbreviation="GFG", full_name="geeksforgeeks")
```

```
The value of abbreviation is GFG
The value of full_name is geeksforgeeks
```

In [ ]:
```

```