

Using the "key" parameter in sort method

In [1]:

```
a = [56,26,48,58,256,46]
```

In [2]:

```
a.sort()
```

In [3]:

```
a
```

Out[3]:

```
[26, 46, 48, 56, 58, 256]
```

In [4]:

```
b = [[5,6],[1,8],[9,2],[0,11],[4,6]]
```

In [5]:

```
b.sort()  
b
```

Out[5]:

```
[[0, 11], [1, 8], [4, 6], [5, 6], [9, 2]]
```

In [6]:

```
def mySort(l):  
    return l[1]
```

In [9]:

```
b.sort(key = mySort)  
b
```

Out[9]:

```
[[9, 2], [4, 6], [5, 6], [1, 8], [0, 11]]
```

In [12]:

```
b.sort(key = lambda x:x[1])  
b
```

Out[12]:

```
[[9, 2], [4, 6], [5, 6], [1, 8], [0, 11]]
```

In [13]:

```
a
```

Out[13]:

```
[26, 46, 48, 56, 58, 256]
```

In [14]:

```
a = [56,26,48,58,256,46]
```

In [15]:

```
sorted(a)
```

Out[15]:

```
[26, 46, 48, 56, 58, 256]
```

In [18]:

```
sorted(b, key=lambda x:x[1])
```

Out[18]:

```
[[9, 2], [4, 6], [5, 6], [1, 8], [0, 11]]
```

In [25]:

```
#i = sorted(range(len(b)), key = lambda x:b[1])  
#i
```

Out[25]:

```
[0, 1, 2, 3, 4]
```

List comprehension in Python

In [26]:

```
# Writing list and the loop in a single line
```

In [27]:

```
[i for i in range(1,10)]
```

Out[27]:

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

In [28]:

```
c = []  
for i in range(1,10):  
    c.append(i)  
c
```

Out[28]:

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

In [29]:

```
# Even numbers using list comprehension
```

In [30]:

```
[j for j in range(1,20) if j%2==0]
```

Out[30]:

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

In [31]:

```
name = ['Yogesh', 'Abhishek', 'Soniya', 'Pooja', 'Mohan']  
[len(i) for i in name]
```

Out[31]:

```
[6, 8, 6, 5, 5]
```

In [32]:

```
[i.upper() for i in name]
```

Out[32]:

```
['YOGESH', 'ABHISHEK', 'SONIYA', 'POOJA', 'MOHAN']
```

In [33]:

```
d = [input() for i in range(3)]
```

```
4  
3  
7
```

In [34]:

```
d
```

Out[34]:

```
['4', '3', '7']
```

In [36]:

```
e = [int(input()) for i in range(3)]
```

3
4
5

In [37]:

```
e
```

Out[37]:

```
[3, 4, 5]
```

In [38]:

```
# above cell is for taking input in new line
```

In [39]:

```
f=[i for i in input().split(' ')]
```

4 5 6 7 8

In [40]:

```
f
```

Out[40]:

```
['4', '5', '6', '7', '8']
```

In [41]:

```
g=[int(i) for i in input().split(' ')]
```

3 4 5 6 7

In [42]:

```
g
```

Out[42]:

```
[3, 4, 5, 6, 7]
```

In [43]:

```
# above cell is taking input in a single line
```

In [44]:

```
[i for i in zip(f,g)]
```

Out[44]:

```
[('4', 3), ('5', 4), ('6', 5), ('7', 6), ('8', 7)]
```

In [45]:

```
[(int(p),q) for p,q in zip(f,g)]
```

Out[45]:

```
[(4, 3), (5, 4), (6, 5), (7, 6), (8, 7)]
```

In [46]:

```
#WAP using list comprehension to generate list of your names characters
```

```
[i for i in "Yogesh"]
```

Out[46]:

```
['Y', 'o', 'g', 'e', 's', 'h']
```

In [47]:

```
['even' if i%2==0 else 'odd' for i in range(10)]
```

Out[47]:

```
['even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd']
```

In [52]:

```
[i for i in range(100) if i%2==0 if i%7 == 0]
```

Out[52]:

```
[0, 14, 28, 42, 56, 70, 84, 98]
```

In [53]:

```
# problem 1
```

In [61]:

```
p = [int(input()) for i in range(5)]
q = [int(input()) for i in range(5)]
```

2
2
2
2
3
2
2
3
3
4

In [62]:

```
p,q
```

Out[62]:

```
([2, 2, 2, 2, 3], [2, 2, 3, 3, 4])
```

In [64]:

```
if len(p)==len(q):
    if all([True if i>=j else False for i,j in zip(p,q)]):
        print("Compatible")
    else:
        print("Incompatible")
else:
    print("Incompatible")
```

Incompatible

In [65]:

```
# Problem no. 2
```

In [66]:

```
n = int(input())
a = [int(i) for i in input().split(' ')]
```

6
7 6 8 16 12 3

In [75]:

```
a = [7, 6, 8, 16, 12, 3]
```

In [76]:

```
for i in range(len(a)-1, 0, -1):  
    a[i-1] = a[i-1]-a[i]  
print(a)
```

[2, 5, 1, 7, 9, 3]

In [77]:

```
a[0],sum(a)
```

Out[77]:

(2, 27)

Dictionary comprehension

In [78]:

```
{i:i**3 for i in range(1,9)}
```

Out[78]:

{1: 1, 2: 8, 3: 27, 4: 64, 5: 125, 6: 216, 7: 343, 8: 512}

In [79]:

```
product = ['Paneer', 'Milk', 'Curd', 'Cheese']  
price = [1000, 70, 85, 200]
```

In [81]:

```
{i:j for i,j in zip(product, price)}
```

Out[81]:

{'Paneer': 1000, 'Milk': 70, 'Curd': 85, 'Cheese': 200}

In [84]:

```
{i:round(j/82,2) for i,j in zip(product,price)}
```

Out[84]:

{'Paneer': 12.2, 'Milk': 0.85, 'Curd': 1.04, 'Cheese': 2.44}

Nested loop comprehension

In [87]:

```
[[x,y] for x in [1,2,3] for y in [4,5,6]]
```

Out[87]:

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

In [88]:

```
eval("".join(str(i) for i in range(1,6)))
```

Out[88]:

120

In [89]:

```
eval("1*2*3*4*5")
```

Out[89]:

120

In [90]:

```
b = [[5,6],[1,8],[9,2],[0,11],[4,6]]
```

In [91]:

```
f = sorted(range(len(b)),key=b.__getitem__)
```

In [92]:

f

Out[92]:

[3, 1, 4, 0, 2]

In [93]:

```
for i in f:  
    print(b[i])
```

[0, 11]

[1, 8]

[4, 6]

[5, 6]

[9, 2]

In [97]:

```
g = sorted(range(len(b)),key=lambda x:b[0])  
g
```

Out[97]:

[0, 1, 2, 3, 4]

In [110]:

```
g=sorted(enumerate(b), key=operator.itemgetter(1))  
g
```

Out[110]:

```
[(3, [0, 11]), (1, [1, 8]), (4, [4, 6]), (0, [5, 6]), (2, [9, 2])]
```

Map and Filter function

In [113]:

```
# Without using loop pointing to each and every element of the List
```

In [114]:

```
# Using function on Lambda to achieve the above
```

In [115]:

```
b = list(range(1,20))  
b
```

Out[115]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [116]:

```
[i**2 for i in b]
```

...

In [117]:

```
def mysq(i):  
    return i*i
```

In [119]:

```
list(map(mysq,b))
```

...

In [121]:

```
list(map(lambda i:i*i, b))
```

Out[121]:

```
[1,  
4,  
9,  
16,  
25,  
36,  
49,  
64,  
81,  
100,  
121,  
144,  
169,  
196,  
225,  
256,  
289,  
324,  
361]
```

In [122]:

```
name
```

Out[122]:

```
['Yogesh', 'Abhishek', 'Soniya', 'Pooja', 'Mohan']
```

In [124]:

```
list(map(len, name))
```

Out[124]:

```
[6, 8, 6, 5, 5]
```

In [125]:

```
Hello = list(map(str, input().split(' ')))
```

```
Yogesh Mohan Abhishek
```

In [126]:

```
Hello
```

Out[126]:

```
['Yogesh', 'Mohan', 'Abhishek']
```

In [127]:

```
Hello = list(map(int, input().split(' ')))
```

1 2 3 4

In [128]:

```
Hello
```

Out[128]:

```
[1, 2, 3, 4]
```

In [129]:

```
#Using map generate tuple of uppercase and Length of name  
def u_l(i):  
    return i.upper(), len(i)  
list(map(u_l, name))
```

Out[129]:

```
[('YOGESH', 6), ('ABHISHEK', 8), ('SONIYA', 6), ('POOJA', 5), ('MOHAN', 5)]
```

In [130]:

```
list(map(lambda i:(i.upper(),len(i)),name))
```

Out[130]:

```
[('YOGESH', 6), ('ABHISHEK', 8), ('SONIYA', 6), ('POOJA', 5), ('MOHAN', 5)]
```

In [131]:

```
d = {'YOGESH': 6, 'ABHISHEK': 8, 'SONIYA': 6, 'POOJA':5, 'MOHAN': 5}
```

In [133]:

```
list(map(len,d))
```

Out[133]:

```
[6, 8, 6, 5, 5]
```

In [134]:

```
list(map(lambda i:i**2, d.values()))
```

Out[134]:

```
[36, 64, 36, 25, 25]
```

In [135]:

```
b
```

Out[135]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [137]:

```
list(filter(lambda i:i%2==0, b))
```

Out[137]:

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

In [138]:

```
name
```

Out[138]:

```
['Yogesh', 'Abhishek', 'Soniya', 'Pooja', 'Mohan']
```

In [139]:

```
def lenovo(i):  
    return 's' not in i
```

In [140]:

```
list(filter(lenovo, name))
```

Out[140]:

```
['Soniya', 'Pooja', 'Mohan']
```

In [141]:

```
list(filter(lambda i:'s' not in i, name))
```

Out[141]:

```
['Soniya', 'Pooja', 'Mohan']
```

In [142]:

```
# filter is taken at the point of bool values
```

In [144]:

```
t = int(input())
for i in range(t):
    n,m = list(map(int,input().split(' ')))
    print(n*m)
```

```
2
3 10
30
1 10
10
```

In [146]:

```
t = int(input())
for i in range(t):
    u = list(map(int,input().split(' ')))
    u.sort()
    print(u[-2])
```

```
3746 267 3876 1457
3746
```

In [149]:

```
t = int(input())
for i in range(t):
    s,c = list(map(int,input().split(' ')))
    if c>=s:
        print(0)
    else:
        print(s-c)
```

```
20 14
6
```

In [150]:

```
t = int(input())
for i in range(t):
    x,y,z = list(map(int,input().split(' ')))
    print(x*4+y*2)
```

```
1
6 6 5
36
```

We need to find how many vowels are there in each name in list "name".

In [154]:

```
name = ['Yogesh', 'Abhishek', 'Soniya', 'Pooja', 'Mohan']
l=[]
for i in name:
    c=0
    for j in i:
        if j.lower() in 'aeiou':
            c += 1
    l.append((i,c))
print(l)
```

```
[('Yogesh', 2), ('Abhishek', 3), ('Soniya', 3), ('Pooja', 3), ('Mohan', 2)]
```

Can we do the same program using list comprehension?

In [156]:

```
def countvowels(s):
    c=0
    for i in s:
        if i.lower() in 'aeiou':
            c+=1
    return c
lc = [(i,countvowels(i)) for i in name]
lc
```

Out[156]:

```
[('Yogesh', 2), ('Abhishek', 3), ('Soniya', 3), ('Pooja', 3), ('Mohan', 2)]
```

In []: