In [3]:

```
l = ['athens', 'heraklion', 'patras']
min(l)
```

Out[3]:

```
'athens'
```

In [2]:

```
def f(x):
    return x[1]
```

In [4]:

```
min(l, key=f)
```

Out[4]:

```
'patras'
```

In [5]:

```
l = [
    ['gene_1', 20],
    ['gene_2', 50],
    ['gene_3', 40],
]
```

In [6]:

```
l[0]
#f(l[0])
```

Out[6]:

```
['gene_1', 20]
```

In [7]:

```
f(['gene_1', 20])
```

Out[7]:

```
20
```

In [8]:

```
f('mitsos')
```

Out[8]:

```
'i'
```

In [9]:

```
min(l, key=f)
```

Out[9]:

```
['gene_1', 20]
```

In [10]:

```
max(l, key=f)
```

Out[10]:

```
['gene_2', 50]
```

In [11]:

```
max(l, key=f)[0]
```

Out[11]:

```
'gene_2'
```

In [12]:

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

Out[12]:

```
1
```

In [13]:

```
l = [
    ['gene_1', 20],
    ['gene_2', 50],
    ['gene_3', 40],
]
max(l, key=f)
```

Out[13]:

```
['gene_2', 50]
```

```
In [14]: max(l, key=f) # --> ['gene_2', 50]

Out[14]: ['gene_2', 50]
```

```
In [15]: ['gene_2', 50][0]

Out[15]: 'gene_2'
```

```
In [16]: max(l, key=f)[0]

Out[16]: 'gene_2'
```

Φτιάξτε μία συνάρτηση η οποοία θα παίρνει μία παράμετρ η οποία θα είναι μία λίστα από strings. Η συνάρτηση θα πρέπει να επιστρέφει το στοιχείο της λίστα το οποίο έχει το μικρότερα αλφαβητικά 2ο γράμμα.

```
In [17]: def f(x):
             return x[1]

         def g(l):
             return min(l, key=f)
```

```
In [18]: g(['alex', 'kostas', 'maria'])

Out[18]: 'maria'
```

Φτιάξτε μία συνάρτηση η οποοία θα παίρνει μία παράμετρ η οποία θα είναι μία λίστα από strings. Η συνάρτηση θα πρέπει να επιστρέφει μία νέα λίστα η οποία θα περιέχει το πλήθος από χαρακτήρες που έχουν τα στοιχεία της λίστας της παραμέτρου.

```
In [19]: fff(['alex', 'kostas', 'maria']) # [4, 5, 5]

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-19-fc3247754b83> in <module>
----> 1 fff(['alex', 'kostas', 'maria']) # [4, 5, 5]

NameError: name 'fff' is not defined
```

```
In [22]: def f2(x):
             return len(x)

         def f1(l):
             return list(map(f2, l))
```

```
In [23]: f1(['alex', 'kostas', 'maria'])

Out[23]: [4, 6, 5]
```

```
In [26]: l1 = list(range(1,101))
```

```
In [28]: all([True, True, True])

Out[28]: True
```

```
In [29]: all([True, True, True, 0])
```

```
Out[29]: False

In [30]: all(['aaa', 'bbb', 'ccc'])

Out[30]: True

In [31]: all(['aaa', 'bbb', '', 'ccc'])

Out[31]: False

In [32]: any([False, False, False])

Out[32]: False

In [33]: any([False, False, 'mitsos', False])

Out[33]: True

In [34]: any([False, False, True, False])

Out[34]: True

In [35]: any([])

Out[35]: False

In [36]: all([])

Out[36]: True

In [37]: cities= ['athens', 'heraklion', 'patras']

In [39]: population = [4_000_000, 200_000, 180_000]

In [40]: list(zip(cities, population))

Out[40]: [('athens', 4000000), ('heraklion', 200000), ('patras', 180000)]

In [41]: all_together = list(zip(cities, population))

In [42]: all_together

Out[42]: [('athens', 4000000), ('heraklion', 200000), ('patras', 180000)]

In [43]: def f(x):
             return x[1]

In [44]: min(all_together, key=f)[0]

Out[44]: 'patras'

In [45]: min(zip(cities, population), key=f)[0]

Out[45]: 'patras'

In [46]: zip(cities, population)
```

```
Out[46]: <zip at 0x7fe95c4544c0>

In [47]: filter(f, cities)

Out[47]: <filter at 0x7fe95c44c4c0>

In [48]: map(f, cities)

Out[48]: <map at 0x7fe95c44c310>

In [49]: b = zip(cities, population)

In [50]: b

Out[50]: <zip at 0x7fe95c4588c0>

In [51]: list(b)

Out[51]: [('athens', 4000000), ('heraklion', 200000), ('patras', 180000)]

In [52]: b

Out[52]: <zip at 0x7fe95c4588c0>

In [54]: b = range(1,1_000_000_001)

In [55]: b = range(1,100_000_000)

In [56]: b

Out[56]: range(1, 100000000)

In [57]: c = list(b)

In [58]: del c

In [59]: cities

Out[59]: ['athens', 'heraklion', 'patras']

In [63]: a = [1,2,3,5]
         b = [6,7,8]
         c = ['a', 'v', 'c']

         list(zip(a,b,c))

Out[63]: [(1, 6, 'a'), (2, 7, 'v'), (3, 8, 'c')]

In [66]: list(zip(a[::3], b))

Out[66]: [(1, 6), (5, 7)]

In [67]: a = [5,6,7,6,5,4,5,6,7,2,8,9,7,6]

In [68]: min(a)
```

```
Out[68]: 2
```

```
In [69]: a.index(min(a))
```

```
Out[69]: 9
```

```
In [70]: list(enumerate(a))
```

```
Out[70]: [(0, 5),
          (1, 6),
          (2, 7),
          (3, 6),
          (4, 5),
          (5, 4),
          (6, 5),
          (7, 6),
          (8, 7),
          (9, 2),
          (10, 8),
          (11, 9),
          (12, 7),
          (13, 6)]
```

```
In [71]: def f(x):
             return x[1]
```

```
In [72]: min(enumerate(a), key=f)
```

```
Out[72]: (9, 2)
```

```
In [74]: min(enumerate(a), key=f)[0]
```

```
Out[74]: 9
```

```
In [77]: l = [6,7,8,9]

         for x in l:
             print ('mitsos')
             print (x)
             print ('kostas')
```

```
mitsos
6
kostas
mitsos
7
kostas
mitsos
8
kostas
mitsos
9
kostas
```

In [78]:

```
for x in l:

    if x in [6,8]:
        continue

    print ('mitsos')
    print (x)
    print ('kostas')
```

```
mitsos
7
kostas
mitsos
9
kostas
```

In [79]:

```
for x in l:

    #print ('mitsos')
    print (x)
    #print ('kostas')

    if x in [6,8]:
        continue
```

```
6
7
8
9
```

In [82]:

```
l = [1,2,3,4,5,6,7,8,9,10]
for x in l:

    #if x <= 5:
    #    print (x)
```

```
1
2
3
4
5
```

In [84]:

```
for x in l:
    a=x
    print (a)
```

```
1
2
3
4
5
6
7
8
9
10
```

In [85]:

```
for x in l:
    a=x
    if x <= 5:
        print (a)
```

```
1
2
```

```
3
4
```

In [87]:

```
for x in l:
    a=x
    print (a)
    if x <= 5:
        print (a)
```

```
1
1
2
2
3
3
4
4
5
5
6
7
8
9
10
```

In [96]:

```
a = list(range(1,101))
def f1(x):
    return [a,x]

b = list(map(f1, a))

def f2(k):

    def f3(x):
        return x%k==0

    return f3
```

In [98]:

```
list(map(f2(b[0][1]), b[0][0]))
```

Out[98]:

```
[True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True,
 True]
```

```
In [89]: a
```

```
Out[89]: [1,
 2,
 3,
 4,
 5,
 6,
 7,
 8,
 9,
 10,
 11,
 12,
 13,
 14,
 15,
 16,
 17,
 18,
 19,
 20,
 21,
 22,
 23,
 24,
 25,
 26,
 27,
 28,
 29,
 30,
 31,
 32,
 33,
 34,
 35,
 36,
 37,
 38,
 39,
 40,
 41,
 42,
 43,
 44,
 45,
 46,
 47,
 48,
 49,
 50,
 51,
 52,
 53,
 54,
 55,
 56,
 57,
 58,
 59,
 60,
 61,
 62,
 63,
 64,
 65,
 66,
 67,
 68,
 69,
 70,
 71,
 72,
 73,
 74,
 75,
 76,
 77,
 78,
 79,
 80,
 81,
 82,
 83,
 84,
 85,
 86,
 87,
 88,
 89,
 90,
 91,
 92,
 93,
 94,
 95,
 96,
 97,
 98,
 99,
```

In [99]:

```
l = [1,2,3,4,5,6,7,8,9,10]
for x in l:
    print (x)
    if x <= 5:
        print (x+10)
```

```
1
11
2
12
```

```
3
13
4
14
5
15
6
7
8
9
10
```

In [100…

```
l = [1,2,3,4,5,6,7,8,9,10]
for x in l:
    print (x)
    if x > 5:
        continue
    print (x+10)
```

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

In [101…

```
l = [12,5,18,8,7,20]

for x in l:
    if x <10:
        continue

    print (x)
```

```
12
18
20
```

In [104…

```
l = [1,2,3,4,5,6,7,8,9,10]
for x in l:
    print (x)

    if x > 5:
        break
```

```
1
2
3
4
5
6
```

In [105…

```
for x in l:
    if x > 5:
        break
    print (x)
```

```
1
2
3
4
5
```

In [106…

```
for x in 'mitsos':
    print (x)
```

```
m
i
t
s
o
s
```

In [107…

```
for x in [[1,2], [6,7], [7,8]]:
    print (x)
```

```
[1, 2]
[6, 7]
[7, 8]
```

In [108…

```
a=3
```

In [109…

```
a,b = [4,5]
```

In [110…

```
a
```

Out[110…

```
4
```

In [111…

```
b
```

Out[111…

```
5
```

In [114…

```
for x,y in [[1,2], [6,7], [7,8]]:
    print (f'{x}+{y}={x+y}')
```

```
1+2=3
6+7=13
7+8=15
```

In [115…

```
l = [2,3,4,5,6,7,8,]
```

In [116…

```
def f(x):
    return x+10

list(map(f, l))
```

Out[116…

```
[12, 13, 14, 15, 16, 17, 18]
```

In [118…

```
k = []
for x in l:
    k.append(x+10)
print (k)
```

```
[12, 13, 14, 15, 16, 17, 18]
```

In [120…

```
k = []
for x in l:
    k += [x+10]
print (k)
```

```
[12, 13, 14, 15, 16, 17, 18]
```

In [123…

```
k = [x+10 for x in l]
print (k)
```

```
[12, 13, 14, 15, 16, 17, 18]
```

In [124…

```
l = [5,6,7,8,9,10,11]
```

In [125…

```
def f(x):
    return sum(x)
```

In [126…

```
f(l)
```

Out[126…

```
56
```

In [127…

```
def f2(x):
    return x+10
f(map(f2, l))
```

Out[127…

```
126
```

In [128…

```
k = []
for x in l:
    k += [x+10]
f(k)
```

Out[128…

```
126
```

In [129…

```
f( [x+10 for x in l] )
```

Out[129…

```
126
```

In [130…

```
[x+10 for x in l] + ['mitsos']
```

Out[130…

```
[15, 16, 17, 18, 19, 20, 21, 'mitsos']
```

In [132…

```
l = [2,3,4,5,6,7,8,9,10]
```

In [133…

```
k = []
for x in l:
    if x>5:
        k.append(x+10)

k
```

Out[133…

```
[16, 17, 18, 19, 20]
```

In [134…

```
[ x+10 for x in l if x>5  ]
```

Out[134…

```
[16, 17, 18, 19, 20]
```

In [135…

```
l
```

Out[135…

```
[2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [138…

```
for x in range(1,11):
    print (f'{x} times 5 is {x*5}')
```

```
1 times 5 is 5
2 times 5 is 10
3 times 5 is 15
4 times 5 is 20
5 times 5 is 25
6 times 5 is 30
7 times 5 is 35
8 times 5 is 40
9 times 5 is 45
10 times 5 is 50
```

In [145…

```
for y in range(1,11):
    for x in range(1,11):
        #print (y,x)
        print (f'{x} times {y} is {x*y}')
```

```
1 times 1 is 1
2 times 1 is 2
3 times 1 is 3
4 times 1 is 4
5 times 1 is 5
6 times 1 is 6
7 times 1 is 7
8 times 1 is 8
9 times 1 is 9
10 times 1 is 10
1 times 2 is 2
2 times 2 is 4
3 times 2 is 6
4 times 2 is 8
5 times 2 is 10
6 times 2 is 12
7 times 2 is 14
8 times 2 is 16
9 times 2 is 18
10 times 2 is 20
1 times 3 is 3
2 times 3 is 6
3 times 3 is 9
4 times 3 is 12
5 times 3 is 15
6 times 3 is 18
7 times 3 is 21
8 times 3 is 24
9 times 3 is 27
10 times 3 is 30
1 times 4 is 4
2 times 4 is 8
3 times 4 is 12
4 times 4 is 16
5 times 4 is 20
6 times 4 is 24
```

```
7 times 4 is 28
8 times 4 is 32
9 times 4 is 36
10 times 4 is 40
1 times 5 is 5
2 times 5 is 10
3 times 5 is 15
4 times 5 is 20
5 times 5 is 25
6 times 5 is 30
7 times 5 is 35
8 times 5 is 40
9 times 5 is 45
10 times 5 is 50
1 times 6 is 6
2 times 6 is 12
3 times 6 is 18
4 times 6 is 24
5 times 6 is 30
6 times 6 is 36
7 times 6 is 42
8 times 6 is 48
9 times 6 is 54
10 times 6 is 60
1 times 7 is 7
2 times 7 is 14
3 times 7 is 21
4 times 7 is 28
5 times 7 is 35
6 times 7 is 42
7 times 7 is 49
8 times 7 is 56
9 times 7 is 63
10 times 7 is 70
1 times 8 is 8
2 times 8 is 16
3 times 8 is 24
4 times 8 is 32
5 times 8 is 40
6 times 8 is 48
7 times 8 is 56
8 times 8 is 64
9 times 8 is 72
10 times 8 is 80
1 times 9 is 9
2 times 9 is 18
3 times 9 is 27
4 times 9 is 36
5 times 9 is 45
6 times 9 is 54
7 times 9 is 63
8 times 9 is 72
9 times 9 is 81
10 times 9 is 90
1 times 10 is 10
2 times 10 is 20
3 times 10 is 30
4 times 10 is 40
5 times 10 is 50
6 times 10 is 60
7 times 10 is 70
8 times 10 is 80
9 times 10 is 90
10 times 10 is 100
```

In [144…

```
a=3
b=543
f'the result is {a**b}'
```

Out[144…

```
'the result is 119355191184927318611704533269955810603983814822853343491062
2011577322329414952429120918519306233415195126333814597800000541985234306704
9171344263624658770848526822548919097505597663169917391029784291289028924390
53882155411807505463915364234260497974226854762 7'
```

In [146…

```
#[ for y in range(1,11) for x in range(1,11) ]
```

In [147…

```
[[x,y] for y in ['a', 'b', 'c'] for x in [5,7] ]
```

Out[147…

```
[[5, 'a'], [7, 'a'], [5, 'b'], [7, 'b'], [5, 'c'], [7, 'c']]
```

In [152…

```
k = [f'{x}X{y}={x*y}' for y in range(1,11) for x in range(1,11)]
```

In [156…

```
#print ('\n'.join(k))
print ('\n'.join([f'{x}X{y}={x*y}' for y in range(1,11) for x in range(1,1
```

```
1X1=1
2X1=2
3X1=3
4X1=4
5X1=5
6X1=6
7X1=7
8X1=8
9X1=9
10X1=10
1X2=2
2X2=4
3X2=6
4X2=8
5X2=10
6X2=12
7X2=14
8X2=16
9X2=18
10X2=20
1X3=3
2X3=6
3X3=9
4X3=12
5X3=15
6X3=18
7X3=21
8X3=24
9X3=27
10X3=30
1X4=4
2X4=8
3X4=12
4X4=16
5X4=20
6X4=24
7X4=28
8X4=32
9X4=36
10X4=40
1X5=5
2X5=10
```

3X5=15
4X5=20
5X5=25
6X5=30
7X5=35
8X5=40
9X5=45
10X5=50
1X6=6
2X6=12
3X6=18
4X6=24
5X6=30
6X6=36
7X6=42
8X6=48
9X6=54
10X6=60
1X7=7
2X7=14
3X7=21
4X7=28
5X7=35
6X7=42
7X7=49
8X7=56
9X7=63
10X7=70
1X8=8
2X8=16
3X8=24
4X8=32
5X8=40
6X8=48
7X8=56
8X8=64
9X8=72
10X8=80
1X9=9
2X9=18
3X9=27
4X9=36
5X9=45
6X9=54
7X9=63
8X9=72
9X9=81
10X9=90
1X10=10
2X10=20
3X10=30
4X10=40
5X10=50
6X10=60
7X10=70
8X10=80
9X10=90
10X10=100

In [163…

```
metritis = 0
for x in range(1,46):
    for y in range(x+1,46):
        for z in range(y+1,46):
            for a in range(z+1,46):
                for b in range(a+1,46):
                    for t in range(1,21):
                        #print (x,y,z,a,b,t)
                        metritis += 1
                        if metritis % 1000000 == 0:
                            print (metritis)
print (metritis)
```

```
1000000
2000000
3000000
4000000
5000000
6000000
7000000
8000000
9000000
10000000
11000000
12000000
13000000
14000000
15000000
16000000
17000000
18000000
19000000
20000000
21000000
22000000
23000000
24000000
24435180
```

In [168…

```
len([
'a'

for x in range(1,46)
    for y in range(x+1,46)
        for z in range(y+1,46)
            for a in range(z+1,46)
                for b in range(a+1,46)
                    for t in range(1,21)
                      if x+y==z
])
```

Out[168…

```
1421420
```

In [169…

```
a=5
a += 1
a
```

Out[169…

```
6
```

In [170…

```
a = 5
a = a+ 1 # a += 1
a
```

Out[170… 6

In [172…

```
metritis = 0

metritis += 1
metritis += 1
metritis += 1
metritis += 1
metritis += 1
metritis += 1


print (metritis)
```

```
6
```

In [173…

```
for x in range(1,6):
    for y in range(10,15):
        print (x,y)
```

```
1 10
1 11
1 12
1 13
1 14
2 10
2 11
2 12
2 13
2 14
3 10
3 11
3 12
3 13
3 14
4 10
4 11
4 12
4 13
4 14
5 10
5 11
5 12
5 13
5 14
```

In [175…

```
list(range(10,15))
```

Out[175… [10, 11, 12, 13, 14]

In [ ]: