In [3]:
```
a = [4,8,5,9,10]
```

In [12]:
```
import random
students = [
    'Tzwrtzina',
    'Andreas',
    'Xristos',
    'Andromaxh',
    'Danah',
    'Antwnia',
    'Aris',
    'Maria',
    'Sofia',
    'Iwanna',
    'Aggelos',
]

def random_student():
    return random.choice(students)
rs = random_student
```

In [ ]:
```

```

In [2]:
```
sorted(a)
```

Out[2]:
```
[4, 5, 8, 9, 10]
```

In [5]:
```
[8,9,10]
```

Out[5]:
```
[8, 9, 10]
```

Sets

In [6]:
```
set( [1,2,1,2,1,2,1,2,3,2,1,3,3,2,1] )
```

Out[6]:
```
{1, 2, 3}
```

In [18]:
```
a = {1,2,3,4}
b = {3,4,5,6}
```

In [20]:
```
a & b
```

Out[20]:
```
{3, 4}
```

In [22]:
```
a | b
```

Out[22]:
```
{1, 2, 3, 4, 5, 6}
```

In [24]:
```
a - b
```

Out[24]:
```
{1, 2}
```

In [25]:
```
b - a
```

Out[25]:
```
{5, 6}
```

In [26]:
```
a
```

Out[26]:
```
{1, 2, 3, 4}
```

In [27]:
```
a.add(5)
```

In [28]:
```
a
```

Out[28]:
```
{1, 2, 3, 4, 5}
```

In [29]:
```
a |= {6}
```

In [30]:
```
a
```

Out[30]:
```
{1, 2, 3, 4, 5, 6}
```

In [31]:
```
a = a | {6}
```

In [32]:
```
a
```

Out[32]:
```
{1, 2, 3, 4, 5, 6}
```

In [33]:
```
len(a)
```

Out[33]:
```
6
```

In [35]:
```
len({})
```

Out[35]:
```
0
```

In [36]:
```
a
```

Out[36]:
```
{1, 2, 3, 4, 5, 6}
```

In [37]:
```
type(a) is set
```

Out[37]:
```
True
```

In [38]:
```
set('s.dkjfhakjdfhalkehfds.fna.sdjgfh`.dkfhjs.dkfgjasgf.ad')
```

Out[38]:
```
{'.', '`', 'a', 'd', 'e', 'f', 'g', 'h', 'j', 'k', 'l', 'n', 's
'}
```

In [39]:
```
set([1,2,5,3,2,3,4,3,2,])
```

Out[39]:
```
{1, 2, 3, 4, 5}
```

In [40]:
```
5 in {1,3,6,8}
```

Out[40]:
```
False
```

In [41]:
```
5 in {1,3,5,6,8}
```

Out[41]:
```
True
```

In [44]:
```
{x%5 for x in range(100)}
```

Out[44]:
```
{0, 1, 2, 3, 4}
```

In [ ]:
```

```

In [15]:
```
a = {3,4,5}
```

In [16]:
```
b = {3:'a', 4:'b', 5:'c'}
```

In [17]:
```
b
```

Out[17]:
```
{3: 'a', 4: 'b', 5: 'c'}
```

In [23]:
```
rs()
```

Out[23]:
```
'Danah'
```

In [48]:
```
import random

l = []
for x in range(50):
    start = random.randint(1,100)
    length = random.randint(5,10)

    l.append((start, start+length))
print (l)
```

```
[(35, 45), (97, 103), (67, 75), (46, 56), (69, 76), (73, 78),
(45, 54), (32, 38), (68, 74), (16, 21), (26, 31), (100, 108),
(5, 11), (67, 74), (3, 9), (67, 74), (91, 96), (47, 56), (66, 7
2), (32, 41), (5, 14), (98, 105), (54, 60), (8, 18), (44, 51),
(71, 77), (52, 60), (13, 21), (53, 63), (27, 34), (39, 49), (2
5, 35), (69, 75), (60, 65), (59, 64), (18, 28), (60, 68), (44,
49), (60, 70), (68, 76), (63, 70), (79, 87), (46, 51), (98, 10
3), (71, 78), (61, 70), (36, 46), (41, 47), (62, 72), (8, 15)]
```

In [64]:
```
l=[(35, 45), (97, 103), (67, 75), (46, 56), (69, 76), (73, 78),
(45, 54), (32, 38), (68, 74), (16, 21), (26, 31), (100, 108),
(5, 11), (67, 74), (3, 9), (67, 74), (91, 96), (47, 56), (66, 7
2), (32, 41), (5, 14), (98, 105), (54, 60), (8, 18), (44, 51),
(71, 77), (52, 60), (13, 21), (53, 63), (27, 34), (39, 49), (25,
35), (69, 75), (60, 65), (59, 64), (18, 28), (60, 68), (44, 49),
(60, 70), (68, 76), (63, 70), (79, 87), (46, 51), (98, 103), (7
1, 78), (61, 70), (36, 46), (41, 47), (62, 72), (8, 15)]
```

In [65]:
```
for x in range(1,100):
    for start, end in l:
        if start<=x<=end:
            break
    else:
        print (x)
```

```
1
2
88
89
90
```

In [66]:
```
for x in range(1,100):

    in_space = False
    for start, end in l:
        if start<=x<=end:
            in_space = True
            break
    if not in_space:
        print (x)
```

```
1
2
88
89
90
```

In [56]:
```
s = set()
for start, end in l:
    s |= set(range(start, end+1))
set(range(1,101)) - s
```

Out[56]:
```
{1, 2, 88, 89, 90}
```

In [58]:
```
[x for start, end in l for x in range(start, end+1)]
```

Out[58]:
```
[35,
 36,
 37,
 38,
 39,
 40,
 41,
 42,
 43,
 44,
 45,
 97,
 98,
 99,
 100,
 101,
 102,
 103,
 67,
 68,
 69,
 70,
 71,
 72,
 73,
 74,
 75,
 46,
 47,
 48,
 49,
 50,
 51,
 52,
 53,
 54,
 55,
 56,
 69,
 70,
 71,
 72,
 73,
 74,
 75,
 76,
 73,
 74,
 75,
 76,
 77,
 78,
 45,
 46,
 47,
 48,
```

```
49,
50,
51,
52,
53,
54,
32,
33,
34,
35,
36,
37,
38,
68,
69,
70,
71,
72,
73,
74,
16,
17,
18,
19,
20,
21,
26,
27,
28,
29,
30,
31,
100,
101,
102,
103,
104,
105,
106,
107,
108,
5,
6,
7,
8,
9,
10,
11,
67,
68,
69,
70,
71,
72,
73,
74,
```

```
3,
4,
5,
6,
7,
8,
9,
67,
68,
69,
70,
71,
72,
73,
74,
91,
92,
93,
94,
95,
96,
47,
48,
49,
50,
51,
52,
53,
54,
55,
56,
66,
67,
68,
69,
70,
71,
72,
32,
33,
34,
35,
36,
37,
38,
39,
40,
41,
5,
6,
7,
8,
9,
10,
11,
12,
```

```
13,
14,
98,
99,
100,
101,
102,
103,
104,
105,
54,
55,
56,
57,
58,
59,
60,
8,
9,
10,
11,
12,
13,
14,
15,
16,
17,
18,
44,
45,
46,
47,
48,
49,
50,
51,
71,
72,
73,
74,
75,
76,
77,
52,
53,
54,
55,
56,
57,
58,
59,
60,
13,
14,
15,
16,
```

```
17,
18,
19,
20,
21,
53,
54,
55,
56,
57,
58,
59,
60,
61,
62,
63,
27,
28,
29,
30,
31,
32,
33,
34,
39,
40,
41,
42,
43,
44,
45,
46,
47,
48,
49,
25,
26,
27,
28,
29,
30,
31,
32,
33,
34,
35,
69,
70,
71,
72,
73,
74,
75,
60,
61,
62,
```

```
63,
64,
65,
59,
60,
61,
62,
63,
64,
18,
19,
20,
21,
22,
23,
24,
25,
26,
27,
28,
60,
61,
62,
63,
64,
65,
66,
67,
68,
44,
45,
46,
47,
48,
49,
60,
61,
62,
63,
64,
65,
66,
67,
68,
69,
70,
68,
69,
70,
71,
72,
73,
74,
75,
76,
63,
```

```
64,
65,
66,
67,
68,
69,
70,
79,
80,
81,
82,
83,
84,
85,
86,
87,
46,
47,
48,
49,
50,
51,
98,
99,
100,
101,
102,
103,
71,
72,
73,
74,
75,
76,
77,
78,
61,
62,
63,
64,
65,
66,
67,
68,
69,
70,
36,
37,
38,
39,
40,
41,
42,
43,
44,
45,
```

```
46,
41,
42,
43,
44,
45,
46,
47,
62,
63,
64,
65,
66,
67,
68,
69,
70,
71,
72,
8,
9,
10,
```

In [67]:
```
l_2 = []
for start, end in l:
    for i in range(start, end+1):
        l_2.append(i)
set(l_2)
```

Out[67]:
```
{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,
91,
92,
93,
94,
95,
96,
97,
98,
99,
100,
101,
102,
103,
104,
105,
106,
107,
108}
```

In [70]:
```
set(range(1,101)) - set(
    [i for start, end in l for i in range(start, end+1)]
)
```

Out[70]:
```
{1, 2, 88, 89, 90}
```

In [ ]:
```

```

In [73]:
```
rs()
```

Out[73]:
```
'Tzwrtzina'
```

In [72]:
```
# Shadowing
```

In [74]:
```
sum([3,5,1])
```

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

In [75]:
```
sum=10
```

In [76]:
```
sum([3,5,1])
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-76-db1f18a1f134> in <module>()
----> 1 sum([3,5,1])

TypeError: 'int' object is not callable
```

sorting

In [81]:
```
a = [5,2,8,7]
```

In [82]:
```
b = sorted(a)
```

In [84]:
```
print (a)
```

```
[5, 2, 8, 7]
```

In [83]:
```
rs()
```

Out[83]:
```
'Aris'
```

In [85]:
```
a.sort() # sorting in-place
```

In [86]:
```
a
```

Out[86]:
```
[2, 5, 7, 8]
```

In [87]:
```
a = [5,2,8,7]
```

In [88]:
```
sorted(a, reverse=True)
```

Out[88]:
```
[8, 7, 5, 2]
```

In [89]:
```
cities = ['Athens', 'Heraklion', 'Thess']
```

In [90]:
```
sorted(cities, key=len)
```

Out[90]:
```
['Thess', 'Athens', 'Heraklion']
```

In [91]:
```
sorted(cities)
```

Out[91]:
```
['Athens', 'Heraklion', 'Thess']
```

In [92]:
```
def my_own(x):
    return x[2]
```

In [93]:
```
sorted(cities, key=my_own)
```

Out[93]:
```
['Thess', 'Athens', 'Heraklion']
```

In [94]:
```
d = {
    'Heraklion' : 200000,
    'Athens': 3500000,
    'Thess': 500000,
}
```

In [95]:
```
rs()
```

Out[95]:
```
'Sofia'
```

In [96]:
```
sorted(d, key=d.get)
```

Out[96]:
```
['Heraklion', 'Thess', 'Athens']
```

In [106]:
```
d
```

Out[106]:
```
{'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [112]:
```
def f(x):
    return d[x]


sorted(d, key=f)
```

Out[112]:
```
['Heraklion', 'Thess', 'Athens']
```

In [114]:
```
d.get('Heraklion')
```

Out[114]:
```
200000
```

In [116]:
```
sorted(d, key=d.get)
```

Out[116]:
```
['Heraklion', 'Thess', 'Athens']
```

In [129]:
```
sorted(d, key=lambda x : d[x], reverse=True) # To be explained
```

Out[129]:
```
['Athens', 'Thess', 'Heraklion']
```

In [119]:
```
d
```

Out[119]:
```
{'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [120]:
```
max(d, key=d.get)
```

Out[120]:
```
'Athens'
```

In [121]:
```
min(d, key=d.get)
```

Out[121]:
```
'Heraklion'
```

In [122]:
```
d
```

Out[122]:
```
{'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [123]:
```
d['Athens']
```

Out[123]:
```
3500000
```

In [124]:
```
d.get('Athens')
```

Out[124]:
```
3500000
```

In [125]:
```
d.get('Athenssss')
```

In [127]:
```
d.get('Athenssss', 'what???')
```

Out[127]:
```
'what???'
```

In [128]:
```
d.get('Athens', 'what???')
```

Out[128]:
```
3500000
```

In [108]:
```
for x in d:
    print (x)
```

```
Heraklion
Athens
Thess
```

In [ ]:
```

```

In [101]:
```
d.get('Heraklion')
```

Out[101]: 200000

In [102]:
```
d.get('Larissa')
```

In [104]:
```
d.get('Heraklion', -1)
```

Out[104]: 200000

In [105]:
```
d.get('Larissa', -1)
```

Out[105]: -1

In [130]:
```
sorted(d, key=d.get, reverse=True)
```

Out[130]: ['Athens', 'Thess', 'Heraklion']

In [131]:
```
sorted(d, key=d.get)
```

Out[131]: ['Heraklion', 'Thess', 'Athens']

In [ ]:
```
def alex_get(d, key, default):
    if key in d:
        return d[key]
    return default
```

In [132]:
```
d
```

Out[132]: {'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}

In [135]:
```
city = 'Larissa'

population = d.get(city, False)

if population:
    print (population)
else:
    print ('Could not find it')
```

Could not find it

In [ ]:
```

```

In [100]:
```
d['Heraklion']
```

Out[100]: 200000

In [103]:
```
d['Larissa']
```

```
---------------------------------------------------------------------
------------
KeyError                                  Traceback (most recen
t call last)
<ipython-input-103-2e76d5636459> in <module>()
----> 1 d['Larissa']

KeyError: 'Larissa'
```

In [141]:
```
a= [random.randint(1,10) for x in range(100)]
```

In [ ]:

In [ ]:

In [144]:
```
sorted((x for x in a if x%2==0), reverse=True)
```

Out[144]:
```
[10,
 10,
 10,
 10,
 10,
 10,
 10,
 10,
 10,
 10,
 8,
 8,
 8,
 8,
 8,
 6,
 6,
 6,
 6,
 6,
 6,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 4,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2,
 2]
```

In [145]:
```
all(  [True, True, True]  )
```

Out[145]:
```
True
```

In [146]:
```
all(  [True, True, False]  )
```

Out[146]:
```
False
```

In [ ]:
```

```

In [147]:
```
any([ False, False, False, True ])
```

Out[147]:
```
True
```

In [148]:
```
any([ False, False, False, False ])
```

Out[148]:
```
False
```

In [154]:
```
N = 102

prime = not any([N%x==0 for x in range(2,101)])
print (prime)
```

Out[154]:
```
False
```

In [157]:
```
N = 102

prime = all([N%x != 0 for x in range(2,101)])
print (prime)
```

```
False
```

Ternary operator

operators: +,-,*,/, **, %, >, >=, <=, &, |, ==, !=

In [158]:
```
3 * 10
```

Out[158]:
```
30
```

In [159]:
```
3  / 10
```

Out[159]:
```
0.3
```

In [161]:
```
* 10
```

```
  File "<ipython-input-161-422cee96d4d9>", line 1
    * 10
      ^
SyntaxError: can't use starred expression here
```

In [162]:
```
10 *
```

```
  File "<ipython-input-162-15947e36ee64>", line 1
    10 *
       ^
SyntaxError: invalid syntax
```

In [166]:
```
-5
```

Out[166]: -5

In [167]:
```
+5
```

Out[167]: 5

In [168]:
```
not False
```

Out[168]: True

Unary operators: +, -, NOT

Dual operators: *, +, /, ... and, or, <, >

Ternary operator

In [170]:
```
b = 5

a = 'ok' if b>3 else 'not ok'

print (a)
```

```
ok
```

In [171]:
```
b = 1

a = 'ok' if b>3 else 'not ok'

print (a)
```

```
not ok
```

In [ ]:
```
if b>3:
    a='ok'
else:
    a='not ok'
```

In [172]:
```
a = 'ok' if b>3 else 'not ok'
```

In [173]:
```
a = input('Insert a value:')
```

```
Insert a value:5
```

In [174]:
```
print (3)
```

```
3
```

del

In [175]:
```
a=3
```

In [176]:
```
del a
```

In [179]:
```
def f():
    a = list(range(10_000_000))
    return 'mitsos'
```

In [180]:
```
f()
```

Out[180]:
```
'mitsos'
```

In [ ]:
```
### Garbage collector
```

In [178]:
```
1_000_000
```

Out[178]:
```
1000000
```

In [181]:
```
a = [3,5,7,9]
```

In [182]:
```
a = a[:2] + a[3:]
```

In [183]:
```
a
```

Out[183]:
```
[3, 5, 9]
```

In [184]:
```
a = [3,5,7,9]
```

In [185]:
```
del a[2]
```

In [186]:
```
a
```

Out[186]:
```
[3, 5, 9]
```

In [187]:
```
d
```

Out[187]:
```
{'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [188]:
```
del d['Athens']
```

In [189]:
```
d
```

Out[189]:
```
{'Heraklion': 200000, 'Thess': 500000}
```

In [190]:
```
d={'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [191]:
```
d = {k:v for k,v in d.items() if k!='Athens'}
```

In [192]:
```
d
```

Out[192]:
```
{'Heraklion': 200000, 'Thess': 500000}
```

assert

In [193]:
```
a = 5

assert a>0

print ('hello')
```

```
hello
```

In [194]:
```
a = -5

assert a>0

print ('hello')
```

```
---------------------------------------------------------------------
------------
AssertionError                            Traceback (most recen
t call last)
<ipython-input-194-a84775ab75cd> in <module>()
      1 a = -5
      2
----> 3 assert a>0
      4
      5 print ('hello')

AssertionError:
```

In [195]:
```
a = -5

assert a>0, "This shouldn't happen"

print ('hello')
```

```
---------------------------------------------------------------------
------------
AssertionError                            Traceback (most recen
t call last)
<ipython-input-195-e42bd29b08a6> in <module>()
      1 a = -5
      2
----> 3 assert a>0, "This shouldn't happen"
      4
      5 print ('hello')

AssertionError: This shouldn't happen
```

lambda functions

In [196]:
```
def f(x):
    return x/2
```

In [202]:
```
f(10)
```

Out[202]: 5.0

In [203]:
```
f = lambda x : x/2
f(10)
```

Out[203]: 5.0

In [205]:
```
d = {'Heraklion': 200000, 'Athens': 3500000, 'Thess': 500000}
```

In [206]:
```
sorted(d, key=d.get)
```

Out[206]: ['Heraklion', 'Thess', 'Athens']

In [207]:
```
sorted(d, key=lambda x : d[x])
```

Out[207]: ['Heraklion', 'Thess', 'Athens']

In [214]:
```
def f(x):
    return d[x]

sorted(d, key=f)
```

Out[214]: ['Heraklion', 'Thess', 'Athens']

In [249]:
```
def g(x):
    return x

g(5)
```

Out[249]: 5

In [233]:
```
g('mitsos')
```

Out[233]: 'mitsos'

In [247]:
```
#g=5
```

In [250]:
```
sorted(d, key=g)
```

Out[250]: ['Athens', 'Heraklion', 'Thess']

In [235]:
```
sorted(d)
```

Out[235]: ['Athens', 'Heraklion', 'Thess']

In [ ]:
```

```

In [226]:
```
def alex_sorted(L, key=g):

    sort_according_list = []
    for x in L:
        sort_according_to_this = key(x)
        #print (x, sort_according_to_this)
        sort_according_list.append(sort_according_to_this)
    #print (sort_according_list)

    enumeration = []
    for index, x in enumerate(sort_according_list):
        #print (index, x)
        #enumeration.append((index, x))
        enumeration.append((x, index))
    #print (enumeration)
    enumeration_sorted = sorted(enumeration)
    #print (enumeration_sorted)
    indexes_of_sorted_values = [x[1] for x in enumeration_sorte
d]
    #print (indexes_of_sorted_values)

    input_as_list = list(L)
    return [input_as_list[i] for i in indexes_of_sorted_values]

alex_sorted(d, f)
```

Out[226]: ['Heraklion', 'Thess', 'Athens']

In [227]:
```
def f(a, b=2):
    return a+b

f(3,5)
```

Out[227]: 8

In [228]:
```
f(3)
```

Out[228]: 5

In [229]:
```
f(3, b=6)
```

Out[229]: 9

In [230]:
```
def f(a,b):
    return a+b

f(3)
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-230-69f0265f11f6> in <module>()
      2     return a+b
      3
----> 4 f(3)

TypeError: f() missing 1 required positional argument: 'b'
```

In [ ]:

In [ ]:

In [199]:
```
a = []
for x in range(10):
    a.append(x/2)
a
```

Out[199]: [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5]

In [201]:
```
[x/2 for x in range(10)] + ['Mitsos']
```

Out[201]: [0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 'Mitsos']

In [243]:
```
def f(x, my_fabulous_function):

    return x + my_fabulous_function(x)
```

In [239]:
```
def fab_a(x):
    return x+10
def fab_b(x):
    return x/2
```

In [245]:
```
f(10, my_fabulous_function=fab_a)
```

Out[245]:
```
30
```

In [246]:
```
f(30, my_fabulous_function=fab_b)
```

Out[246]:
```
45.0
```

In [251]:
```
f(30, my_fabulous_function=15)
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-251-30f2de8dd48b> in <module>()
----> 1 f(30, my_fabulous_function=15)

<ipython-input-243-69204fcac5f5> in f(x, my_fabulous_function)
      3 def f(x, my_fabulous_function):
      4
----> 5     return x + my_fabulous_function(x)

TypeError: 'int' object is not callable
```

Φτιάξτε μία συνάρτηση με το όνομα f η οποία να επιστρέφει μία συνάρτηση η οποία να επιτρέφει μία συνάρτηση η οποία να επιστρέφει μία λίστα της οποίας το 2ο στοιχείο να είναι μία συνάρτηση η οποία να επιστρέφει το string 'mitsos'. Θα πρέπει δηλαδή να μπορώ να κάνω:

```
f()()()[1]()
```

In [255]:
```
def f():

    def g():

        def h():

            def l():
                return 'Mitsos'
            return [55, l]

        return h


    return g
```

In [256]:
```
f()()()[1]()
```

Out[256]:
```
'Mitsos'
```

In [264]:
```
def l():
    return 'mitsos'

def h():
    return [1, l]

def g():
    return h

def f():
    return g
```

In [265]:
```
f()()()[1]()
```

Out[265]:
```
'mitsos'
```

In [266]:
```
def f1():
    return 'Hello'
def f2():
    return 'world'
```

In [267]:
```
l = [f1, f2]
```

In [270]:
```
l[0]()
```

Out[270]:
```
'Hello'
```

In [271]:
```
l[1]()
```

Out[271]:
```
'world'
```

In [269]:
```
rs()
```

Out[269]:
```
'Tzwrtzina'
```

In [1]:
```
a = [31, -28, 14, -12, -4, 44, 47, 2, -48, -5, -43, 32, 0, -4, 2
4, -46, -12, 38, -38, -27, -23, -26, 10, 42, 26, -20, -43, -50,
2, 42, 32, 17, -33, 5, 42, 28, 2, 12, 9, -33, 22, 10, 3, 34, 12,
17, 21, 17, 24, 22, 21, -35, 33, 12, -43, 49, -17, 3, -2,
-25, -29, -35, -26, -25, -22, -33, 10, 26, -41, 29, 6, -10, 15,
-28, -23, -35, -1, -16, 24, -45, -50, -17, 20, 12, -32, 48,
-48, 2, -41, 4, 5, 29, -36, -46, -6, -17, -18, 16, 42, 42]
```

In [281]:
```
len(a)
```

Out[281]:
```
100
```

In [2]:
```
sum
```

Out[2]:
```
<function sum(iterable, start=0, /)>
```

In [17]:
```
l = []
for index, element in enumerate(a):
    #if index == 0:
    #    continue
    #print (a[:index])
    s = sum(a[:index])
    l.append( (s, index ) )
    #print (l)
#print (l)
print (max(l)[1])
#print (l.index(max(l)))
```

```
56
```

In [ ]:
```

```

In [18]:
```
max((sum(a[:index]), index ) for index, element in enumerate
(a))[1]
```

Out[18]:
```
56
```

In [19]:
```
max((sum(a[:index]), index ) for index in range(len(a)))[1]
```

Out[19]:
```
56
```

In [ ]:
```

```

In [15]:
```
sum(a[:0])
```

Out[15]:
```
0
```

In [25]:
```
L = []
for i in range(len(a)):
    for j in range(i+1, len(a)):
        s = sum(a[i:j+1])
        L.append( (s, (i,j)))

print (max(L))
```

```
(344, (28, 55))
```

In [ ]:
```

```

In [ ]:
```

```