In [7]:
```
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 [8]:
```
rs()
```

Out[8]:
```
'Aris'
```

In [9]:
```
for i in range(1,11):

    mitsos_break = False
    for j in range(1,11):
        print (i,j, '=', i*j)
        if j>=5:
            mitsos_break = True
            break

    print (mitsos_break)

    if i>=5:
        break
```

```
1 1 = 1
1 2 = 2
1 3 = 3
1 4 = 4
1 5 = 5
2 1 = 2
2 2 = 4
2 3 = 6
2 4 = 8
2 5 = 10
3 1 = 3
3 2 = 6
3 3 = 9
3 4 = 12
3 5 = 15
4 1 = 4
4 2 = 8
4 3 = 12
4 4 = 16
4 5 = 20
5 1 = 5
5 2 = 10
5 3 = 15
5 4 = 20
5 5 = 25
```

In [19]:
```
### Γράψε κώδικα να σου τυπώνει True/False ανάλογα
### με το αν υπάρχει νούμερο σε ένα string ή όχι

a = 'zldkfjghldskj6hgrldk5jfgasdlkfjas;or99uhgasfjk'
#a = 'ggg'

it_has_digits = False

for x in a:
    if x in '0123456789':
        it_has_no_digits = True
        break

print (it_has_no_digits)
```

```
True
```

In [15]:
```
rs()
```

Out[15]:
```
'Danah'
```

In [20]:
```
a = 'zldkfjghldskj6hgrldk5jfgasdlkfjas;or99uhgasfjk'
#a = 'ggg'

#it_has_digits = False

for x in a:
    if x in '0123456789':
        print (True)
        break
    else:
        print (False)
        break
```

```
False
```

In [21]:
```
i=10

if i%2 == 0:
    print ('even')
else:
    print ('odd')
```

```
even
```

In [23]:
```
for i in range(1,10):
    print (i)
    if i>5:
        break
else:
    print ('No break!')
```

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

List comprehensions

In [26]:
```
a = [3,5,4,6,8,9,8,7,6,5,4,3,4,5,7,8,9,1]

b = []

for i in a:
    if i%2 == 0:
        b.append(i)

print (b)
```

```
[4, 6, 8, 8, 6, 4, 4, 8]
```

In [27]:
```
b = [i for i in a if i%2 == 0] # List comprehension

print (b)
```

```
[4, 6, 8, 8, 6, 4, 4, 8]
```

In [24]:
```
rs()
```

Out[24]:
```
'Tzwrtzina'
```

In [28]:
```
a = [3,5,4,6,8,9,8,7,6,5,4,3,4,5,7,8,9,1]

b = []

for i in a:
    if i%2 == 0:
        b.append(i*3)

print (b)
```

```
[12, 18, 24, 24, 18, 12, 12, 24]
```

In [30]:
```
b = [i*3 for i in a if i%2 == 0] # List comprehension

print (b)
```

```
[12, 18, 24, 24, 18, 12, 12, 24]
```

In [31]:
```
# Ποιο είναι το άθροισμο των μονών στοιχείων της;
a = [3,5,4,6,8,9,8,7,6,5,4,3,4,5,7,8,9,1]

s = 0
for x in a:
    if x % 2 == 1:
        s+=x
print (s)
```

```
54
```

In [33]:
```
sum([x for x in a if x%2==1])
```

Out[33]:
```
54
```

In [122]:
```
# Πάρε όλους του δυνατούς συνδοιασμούς των λιστών:
a = [1, 2, 3]
b = ['a', 'b', 'c']

# 1a 1b 1c 2a 2b 2c 3a 3b 3c

c = []
for i in a:
    for j in b:
        c.append(str(i)+j)

print (c)
```

```
['1a', '1b', '1c', '2a', '2b', '2c', '3a', '3b', '3c']
```

In [123]:
```
from itertools import product

for x,y in product(a,b):
    print (str(x) + y )
```

```
1a
1b
1c
2a
2b
2c
3a
3b
3c
```

In [39]:
```
c = [str(i)+j for i in a for j in b]
print (c)
```

```
['1a', '1b', '1c', '2a', '2b', '2c', '3a', '3b', '3c']
```

In [ ]:
```

```

In [34]:
```
rs()
```

Out[34]:
```
'Andromaxh'
```

Για να γίνει ένα κέικ κάποιος μπορεί να επιλέξει μία από 6 διαφορετικές μάρκες βούτηρο. Οι τιμές τους είναι:

```
[82, 88, 88, 71, 79, 74]
```

Πρέπει επίσης να επιλέξει μία από 8 μάρκες αυγά. Οι τιμές τους είναι:

```
[73, 91, 82, 98, 95, 90, 70, 73]
```

Πρέπει επίσης να επιλέξει μία από 5 μάρκες γάλα. Οι τιμές τους είναι:

```
[97, 90, 89, 81, 99]
```

Ποιος είναι ο μέσος όρος όλων των δυνατών τιμών που μπορεί να κοστίσει ένα κέικ;

In [125]:
```
bout =  [82, 88, 88, 71, 79, 74]
eggs = [73, 91, 82, 98, 95, 90, 70, 73]
milk = [97, 90, 89, 81, 99]

a = []
for i in bout:
    for j in eggs:
        for k in milk:
            a.append(i + j + k)

#print (a)

sum(a)/len(a)
```

Out[125]:
```
255.53333333333333
```

In [126]:
```
a = [b+e+m for b,e,m in product(bout, eggs, milk)]
print (sum(a)/len(a))
```

```
255.53333333333333
```

In [ ]:
```

```

In [51]:
```
a = [i + j + k for i in bout for j in eggs for k in milk]
print (sum(a)/len(a))
```

```
255.53333333333333
```

In [93]:
```
a = [3,5,4,6,8,9,8,7,6,5,4,3,4,5,7,8,9,1]

# Φτιάξε μία λίστα που να έχει όλα τα δυνατά ζευγάρια της λίστας a
a
len(a)

c = 0
b = []
for i in range(18):
    for j in range (i+1, 18):
        #print (i,j)
        #print (a[i], a[j])
        b.append(  (abs(a[i]-a[j]), a[i], a[j]) )
        c += 1

#print (c)
#print (b)
print (min(b))
```

```
(0, 3, 3)
```

In [94]:
```
a = ['Mitsos', 'Kwstas', 'Elenh']
list(enumerate(a))
```

Out[94]:
```
[(0, 'Mitsos'), (1, 'Kwstas'), (2, 'Elenh')]
```

In [ ]:
```

```

In [106]:
```
a = ['Mitsos', 'Kwstas', 'Elenh']


for x in enumerate(a):
    #print (x)

    #print (x[0])

    #print ( a[x[0]:] )

    #print ( a[x[0]+1:] )

    for y in a[x[0]+1:]:
        print (x[1], y)
```

```
Mitsos Kwstas
Mitsos Elenh
Kwstas Elenh
```

In [107]:
```
a = [('Heraklion', 200000), ('Thess', 500000), ('Athens', 400000)]
```

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

```
('Heraklion', 200000)
('Thess', 500000)
('Athens', 4000000)
```

In [111]:
```
for city, population in a:
    #print (city)
    #print (population)
    print (city, population)
```

```
Heraklion 200000
Thess 500000
Athens 4000000
```

In [112]:
```
a = [
     ('Heraklion', 200000, 'A'),
     ('Thess', 500000, 'B'),
    ('Athens', 4000000, 'C')]

for city, population, letter in a:
    print (city, population, letter)
```

```
Heraklion 200000 A
Thess 500000 B
Athens 4000000 C
```

In [113]:
```
a = ['Mitsos', 'Kwstas', 'Elenh']

for index, element_1 in enumerate(a):
    for element_2 in a[index+1:]:
        print (element_1, element_2)
```

```
Mitsos Kwstas
Mitsos Elenh
Kwstas Elenh
```

In [114]:
```
import random
```

In [116]:
```
from itertools import combinations

for x,y in combinations(a, 2):
    print (x,y)
```

```
Mitsos Kwstas
Mitsos Elenh
Kwstas Elenh
```

In [119]:
```
c = 0
for x in combinations(range(49),6):
    c+= 1
print (c)
```

```
13983816
```

In [121]:
```
sum(1 for x in combinations(range(49),6))
```

Out[121]:
```
13983816
```

In [ ]:
```

```

In [87]:
```
a = ['Mitsos', 'Kwstas', 'Elenh']
list(enumerate(a))
```

Out[87]:
```
[(0, 'Mitsos'), (1, 'Kwstas'), (2, 'Elenh')]
```

In [91]:
```
for x in enumerate(a):
    print (x[1])

```

```
Mitsos
Kwstas
Elenh
```

In [ ]:
```

```

In [ ]:
```

```

In [83]:
```
min((abs(a[i]-a[j]), a[i], a[j])
    for i in range(18) for j in range (i+1, 18))
```

Out[83]:
```
(0, 3, 3)
```

In [ ]:
```

```

In [74]:
```
#b = [(a[i], a[j]) for i in range(18) for j in range (i+1, 18)]

b = min([abs(a[i] - a[j]) for i in range(18) for j in range (i+
1, 18)])
```

In [80]:
```
min([5,2,1,3,6,4,1])
```

Out[80]:
```
1
```

In [81]:
```
(0,3,3) < (0,4,4)
```

Out[81]:
```
True
```

In [ ]:

```
```

In [75]:

```
b
```

Out[75]:

```
0
```

In [71]:
```
b
```

Out[71]:
```
[(3, 5),
 (3, 4),
 (3, 6),
 (3, 8),
 (3, 9),
 (3, 8),
 (3, 7),
 (3, 6),
 (3, 5),
 (3, 4),
 (3, 3),
 (3, 4),
 (3, 5),
 (3, 7),
 (3, 8),
 (3, 9),
 (3, 1),
 (5, 4),
 (5, 6),
 (5, 8),
 (5, 9),
 (5, 8),
 (5, 7),
 (5, 6),
 (5, 5),
 (5, 4),
 (5, 3),
 (5, 4),
 (5, 5),
 (5, 7),
 (5, 8),
 (5, 9),
 (5, 1),
 (4, 6),
 (4, 8),
 (4, 9),
 (4, 8),
 (4, 7),
 (4, 6),
 (4, 5),
 (4, 4),
 (4, 3),
 (4, 4),
 (4, 5),
 (4, 7),
 (4, 8),
 (4, 9),
 (4, 1),
 (6, 8),
 (6, 9),
 (6, 8),
 (6, 7),
 (6, 6),
 (6, 5),
 (6, 4),
 (6, 3),
```

```
(6, 4),
(6, 5),
(6, 7),
(6, 8),
(6, 9),
(6, 1),
(8, 9),
(8, 8),
(8, 7),
(8, 6),
(8, 5),
(8, 4),
(8, 3),
(8, 4),
(8, 5),
(8, 7),
(8, 8),
(8, 9),
(8, 1),
(9, 8),
(9, 7),
(9, 6),
(9, 5),
(9, 4),
(9, 3),
(9, 4),
(9, 5),
(9, 7),
(9, 8),
(9, 9),
(9, 1),
(8, 7),
(8, 6),
(8, 5),
(8, 4),
(8, 3),
(8, 4),
(8, 5),
(8, 7),
(8, 8),
(8, 9),
(8, 1),
(7, 6),
(7, 5),
(7, 4),
(7, 3),
(7, 4),
(7, 5),
(7, 7),
(7, 8),
(7, 9),
(7, 1),
(6, 5),
(6, 4),
(6, 3),
(6, 4),
```

```
(6, 5),
(6, 7),
(6, 8),
(6, 9),
(6, 1),
(5, 4),
(5, 3),
(5, 4),
(5, 5),
(5, 7),
(5, 8),
(5, 9),
(5, 1),
(4, 3),
(4, 4),
(4, 5),
(4, 7),
(4, 8),
(4, 9),
(4, 1),
(3, 4),
(3, 5),
(3, 7),
(3, 8),
(3, 9),
(3, 1),
(4, 5),
(4, 7),
(4, 8),
(4, 9),
(4, 1),
(5, 7),
(5, 8),
(5, 9),
(5, 1),
(7, 8),
(7, 9),
(7, 1),
(8, 9),
(8, 1),
```

In [ ]:

In [59]:
```
N = len(a)
N*(N-1)/2
```

Out[59]: 153.0

In [84]:
```
sum(range(1,18))
```

Out[84]: 153

In [127]:
```
from collections import Counter
```

In [128]:
```
Counter('slkdfjghsldkfjghsldkfjghsldkjfghlsdkjfghlsdkjfghsldkjgh
')
```

Out[128]:
```
Counter({'s': 7, 'l': 7, 'k': 7, 'd': 7, 'f': 6, 'j': 7, 'g':
7, 'h': 7})
```

In [129]:
```
Counter('mitsos') + Counter('kwstas')
```

Out[129]:
```
Counter({'m': 1, 'i': 1, 't': 2, 's': 4, 'o': 1, 'k': 1, 'w':
1, 'a': 1})
```

In [145]:
```
for x in enumerate([1,2,3]):
    print (x, type(x))
```

```
(0, 1) <class 'tuple'>
(1, 2) <class 'tuple'>
(2, 3) <class 'tuple'>
```

In [131]:
```
type((4,5,6))
```

Out[131]:
```
tuple
```

In [132]:
```
class MITSOS:

    def f():
        return 3
```

In [133]:
```
a = MITSOS()
```

In [136]:
```
type(a)
```

Out[136]:
```
__main__.MITSOS
```

In [141]:
```
a = (3)
type(a)
```

Out[141]:
```
int
```

In [ ]:
```

```

In [139]:
```
a = (3,)
```

In [140]:
```
type(a)
```

Out[140]:
```
tuple
```

In [147]:
```
a = ['Heraklion', 'Athens', 'Thess']
b = [200000, 4000000, 500000]
c = ['A', 'B', 'C']


list(zip(a,b,c))
```

Out[147]:
```
[('Heraklion', 200000, 'A'), ('Athens', 4000000, 'B'), ('Thess
', 500000, 'C')]
```

In [148]:
```
for k,l,m in zip(a,b,c):
    print (k,l,m)
```

```
Heraklion 200000 A
Athens 4000000 B
Thess 500000 C
```

In [149]:
```
a
```

Out[149]:
```
['Heraklion', 'Athens', 'Thess']
```

In [150]:
```
list(zip(range(len(a)), a))
```

Out[150]:
```
[(0, 'Heraklion'), (1, 'Athens'), (2, 'Thess')]
```

In [151]:
```
a
```

Out[151]:
```
['Heraklion', 'Athens', 'Thess']
```

In [152]:
```
b
```

Out[152]:
```
[200000, 4000000, 500000]
```

In [154]:
```
[(element, b[index]) for index, element in enumerate(a)]
```

Out[154]:
```
[('Heraklion', 200000), ('Athens', 4000000), ('Thess', 500000)]
```

Dictionaries

In [155]:
```
a = ['mitsos', 'kostas', 'elenh']
```

In [157]:
```
a[2]
```

Out[157]:
```
'elenh'
```

In [158]:
```
a['name']
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-158-e49fd23c94fb> in <module>()
----> 1 a['name']

TypeError: list indices must be integers or slices, not str
```

In [159]:
```
person = {
    'name': 'Elenh',
    'age': 20,
    'department': 'biology'
}
```

In [160]:
```
person['name']
```

Out[160]:
```
'Elenh'
```

In [161]:
```
person['age']
```

Out[161]:
```
20
```

In [162]:
```
persons = [
    {
        'name': 'Elenh',
        'age': 20,
        'department': 'biology'
    },
    {
        'name': 'Kwsta',
        'age': 40,
        'department': 'CS'
    },

]
```

In [164]:
```
persons[0]
```

Out[164]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [165]:
```
persons[1]
```

Out[165]:
```
{'name': 'Kwsta', 'age': 40, 'department': 'CS'}
```

In [167]:
```
persons[1]['age']
```

Out[167]:
```
40
```

In [171]:
```
persons[1]['age'] = 50
```

In [ ]:
```
```

In [172]:
```
persons[1]
```

Out[172]:
```
{'name': 'Kwsta', 'age': 50, 'department': 'CS'}
```

In [173]:
```
person
```

Out[173]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [174]:
```
for x in person:
    print (x)
```

```
name
age
department
```

In [175]:
```
for x in person:
    #print (x)
    print (person[x])
```

```
Elenh
20
biology
```

In [176]:
```
for x in person:
    #print (x)
    #print (person[x])
    print (x, '-->', person[x])
```

```
name --> Elenh
age --> 20
department --> biology
```

In [177]:
```
list(person.values())
```

Out[177]:
```
['Elenh', 20, 'biology']
```

In [178]:
```
list(person.keys())
```

Out[178]:
```
['name', 'age', 'department']
```

In [182]:
```
a = [
    ('name','Elenh'),
    ('age', 20),
    ('department', 'biology'),
]

#dict(a,b)
```

In [184]:
```
a
```

Out[184]:
```
[('name', 'Elenh'), ('age', 20), ('department', 'biology')]
```

In [185]:
```
dict(a)
```

Out[185]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [186]:
```
person
```

Out[186]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [187]:
```
dict(zip(person.keys(), person.values())) # ==> person
```

Out[187]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [188]:
```
person = {
    'name': 'mitsos',
    'grades': [1,2,3,4,5]
}
```

In [190]:
```
person['grades'][-1]
```

Out[190]:
```
5
```

In [191]:
```
len(person)
```

Out[191]:
```
2
```

In [192]:
```
len({})
```

Out[192]:
```
0
```

In [193]:
```
person
```

Out[193]:
```
{'name': 'mitsos', 'grades': [1, 2, 3, 4, 5]}
```

In [194]:
```
person = {'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [195]:
```
list(person.keys())
```

Out[195]:
```
['name', 'age', 'department']
```

In [196]:
```
list(person.values())
```

Out[196]:
```
['Elenh', 20, 'biology']
```

In [197]:
```
list(person.items())
```

Out[197]:
```
[('name', 'Elenh'), ('age', 20), ('department', 'biology')]
```

In [198]:
```
dict(person.items()) # --> person
```

Out[198]:
```
{'name': 'Elenh', 'age': 20, 'department': 'biology'}
```

In [199]:
```
for a,b in person.items():
    print (a, '-->', b)
```

```
name --> Elenh
age --> 20
department --> biology
```

Dictionary comprehensions

In [200]:
```
[x/2 for x in range(10)]
```

Out[200]:
```
[0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5]
```

In [212]:
```
a = { x:x/2 for x in range(10) }
print (a)
```

```
{0: 0.0, 1: 0.5, 2: 1.0, 3: 1.5, 4: 2.0, 5: 2.5, 6: 3.0, 7: 3.
5, 8: 4.0, 9: 4.5}
```

In [203]:
```
a[4]
```

Out[203]:
```
2.0
```

In [213]:
```
a['Misos'] = 23445235
```

In [206]:
```
a
```

Out[206]:
```
{0: 0.0,
 1: 0.5,
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235}
```

In [214]:
```
a[(5,'t')] = 'what?'
```

In [208]:
```
a
```

Out[208]:
```
{0: 0.0,
 1: 0.5,
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235,
 (5, 't'): 'what?'}
```

In [209]:
```
a[[5,4]] = 5
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-209-49bd4e98bb23> in <module>()
----> 1 a[[5,4]] = 5

TypeError: unhashable type: 'list'
```

In [210]:
```
#a = 'asdfasdfasdf'
#a[4] = 'd'
```

```
---------------------------------------------------------------------
------------
TypeError                                 Traceback (most recen
t call last)
<ipython-input-210-7b736cec99f3> in <module>()
      1 a = 'asdfasdfasdf'
----> 2 a[4] = 'd'

TypeError: 'str' object does not support item assignment
```

In [215]:
```
a[6.5] = 8.9
a
```

Out[215]:
```
{0: 0.0,
 1: 0.5,
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235,
 (5, 't'): 'what?',
 6.5: 8.9}
```

In [216]:
```
a[True] = 'hello'
```

In [217]:
```
a
```

Out[217]:
```
{0: 0.0,
 1: 'hello',
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235,
 (5, 't'): 'what?',
 6.5: 8.9}
```

In [218]:
```
True
```

Out[218]: True

In [219]:
```
1
```

Out[219]: 1

In [220]:
```
False
```

Out[220]: False

In [221]:
```
0
```

Out[221]: 0

In [222]:
```
True + True
```

Out[222]:
```
2
```

In [224]:
```
#age = 20
age = 3
b = 5 * (age>10)
print (b)
```

```
0
```

In [225]:
```
def f(x):
    return x/2
```

In [226]:
```
a
```

Out[226]:
```
{0: 0.0,
 1: 'hello',
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235,
 (5, 't'): 'what?',
 6.5: 8.9}
```

In [227]:
```
a['p'] = f
```

In [228]:
```
a
```

Out[228]:
```
{0: 0.0,
 1: 'hello',
 2: 1.0,
 3: 1.5,
 4: 2.0,
 5: 2.5,
 6: 3.0,
 7: 3.5,
 8: 4.0,
 9: 4.5,
 'Misos': 23445235,
 (5, 't'): 'what?',
 6.5: 8.9,
 'p': <function __main__.f(x)>}
```

In [229]:
```
a['p'](10)
```

Out[229]:
```
5.0
```

Άσκηση 2

Η χρήση list comprehension είναι προτεινόμενη

Όλα τα ζευγάρια κ,λ θετικών ακεραίων που είναι μικρότεροι ή ίσοι με το 10 όπου το λ διαιρεί ακριβώς το κ είναι:

```
(4, 2), (6, 2), (6, 3), (8, 2), (8, 4), (9, 3), (10, 2), (10, 5)
```

Το άθροισμα των διαφορών αυτώ των ζευγαριών είναι:

```
(4 - 2) + (6 - 2) + (6 - 3) + (8 - 2) + (8 - 4) + (9 - 3) + (10 - 2)
+ (10 - 5) = 38
```

Ποιο είναι το αντίστοιχο άθροισμα διαφορών αν πάρουμε όλα τα ζευγάρια κ,λ που είναι μικρότεροι η ίσοι με το 1000;

In [236]:
```
s = 0
for x in range(2,11):
    for y in range(x+1, 11):
        if y%x == 0:

            s += y-x

print (s)
```

```
38
```

In [237]:
```
sum(y-x for x in range(2,11) for y in range(x+1, 11) if y%x ==
0)
```

Out[237]:
```
38
```

In [238]:
```
sum(y-x for x in range(2,1001) for y in range(x+1, 1001) if y%x
== 0)
```

Out[238]:
```
2465073
```

In [239]:
```
def f(x):
    return (x/2)
```

In [240]:
```
f(
    sum(y-x for x in range(2,1001) for y in range(x+1, 1001) if
y%x == 0)
)
```

Out[240]:
```
1232536.5
```

In [241]:
```
a= sum(y-x for x in range(2,1001) for y in range(x+1, 1001) if
y%x == 0)
print (a)
```

```
2465073
```

In [ ]:

In [242]:
```
L= [1.3, 1.6, 7.5]



f( [int(x) for x in L] )
```

In [247]:
```
a = [('name', 'kvstas'), ('gender', 'boy'), ('age', 20)]
```

In [248]:
```
{key:value for key, value in a}
```

Out[248]:
```
{'name': 'kvstas', 'gender': 'boy', 'age': 20}
```

In [ ]:

In [ ]:

In [ ]:

In [230]:
```
rs()
```

Out[230]:
```
'Antwnia'
```

In [ ]:

In [ ]:

In [156]:
```
rs()
```

Out[156]:
```
'Aggelos'
```

In [ ]: