

# Μαθήματα Python



Online μεταγλωττιστές

<https://editor.raspberrypi.org/en/>

[https://www.onlinegdb.com/online\\_python\\_compiler](https://www.onlinegdb.com/online_python_compiler)

<https://www.programiz.com/python-programming/online-compiler/>

[https://rextester.com/l/python3\\_online\\_compiler](https://rextester.com/l/python3_online_compiler)

[https://www.w3schools.com/python/trypython.asp?filename=demo\\_default](https://www.w3schools.com/python/trypython.asp?filename=demo_default)

<http://www.pythontutor.com/visualize.html>

```
1 # Python Operators
2 x = 5
3 y = 2
4 z = x + y
5 print(str(x) + "+" + str(y) + "=" + str(z))
6 z = x - y
7 print(str(x) + "-" + str(y) + "=" + str(z))
8 z = x * y
9 print(str(x) + "*" + str(y) + "=" + str(z))
10 z = x / y
11 print(str(x) + "/" + str(y) + "=" + str(z))
12 z = x ** y
13 print(str(x) + "**" + str(y) + "=" + str(z))
14 z = x % y
15 print(str(x) + "%" + str(y) + "=" + str(z))
16 x = "Trito";
17 y = " Lykeio";
18 z = x + y;
19 print(x + " + " + y + " = " + z)
```

```
5+2=7
5-2=3
5*2=10
5/2=2.5
5**2=25
5%2=1
Trito + Lykeio = Trito Lykeio
```

```
1 #String-Methods01
2 print("String-Methods01")
3 print("-----\n")
4 txt = "hello world"
5 print("capitalize: " + "hello world => " + txt.capitalize())
6 txt = "HELLO WORLD"
7 print("casefold: " + "HELLO WORLD => " + txt.casefold())
8 # count()
9 txt = "I love apples, apple are my favorite fruit"
10 txt2 = "apple"
11 x = txt.count(txt2)
12 print("txt = " + txt)
13 print("txt2 = " + txt2)
14 print("count:" + "txt2 appears in txt " + str(x) + " times")
```

```
String-Methods01
```

```
-----
capitalize: hello world => Hello world
casefold: HELLO WORLD => hello world
txt = I love apples, apple are my favorite fruit
txt2 = apple
count:txt2 appears in txt 2 times
```

```
1 #String-Methods02
2 print("String-Methods02")
3 #endswith()
4 txt = "Hello, welcome to my world"
5 txt2 = "world"
6 x = txt.endswith(txt2)
7 print("txt = " + txt)
8 print("txt2 = " + txt2)
9 print("endswith():" + "txt ends with txt2 = " + str(x))
10 #len()
11 x = len(txt)
12 print("len():" + "txt length is = " + str(x))
13 #find()
14 txt2 = "welcome"
15 print("find():" + "txt2 is found at txt at = " + str(txt.find(txt2)))
```

```
String-Methods02
txt = Hello, welcome to my world
txt2 = world
endswith():txt ends with txt2 = True
len():txt length is = 26
find():txt2 is found at txt at = 7
```

```
1 #String-Methods03
2 print("String-Methods03")
3 # replace()
4 txt = "I like bananas"
5 txt2 = "apples"
6 x = txt.replace("bananas", txt2)
7 print("txt = " + txt)
8 print("txt2 = " + txt2)
9 print("replace():" + "txt replaced with txt2 = " + x)
10 # upper()
11 x = txt.upper()
12 print("upper():" + "txt to upper case = " + x)
13 txt2 = "I like" # startswith()
14 print("txt2 = " + txt2)
15 print("startswith():" + "txt startswith txt2" + str(txt.startswith(txt2)))
```

```
String-Methods03
txt = I like bananas
txt2 = apples
replace():txt replaced with txt2 = I like apples
upper():txt to upper case = I LIKE BANANAS
txt2 = I like
startswith():txt startswith txt2True
```

```
1 # Math-Methods01
2 import math
3
4 print ("math.sqrt(81) = " + str(math.sqrt(81)))
5 print ("abs(-18) = " + str(abs(-18)))
6 print ("round(7.3) = " + str(round(7.3)))
7 print ("round(7.8) = " + str(round(7.8)))
8 print ("pi = " + str(math.pi))
9
10 import random
11 x = random.random()
12 print("random number between 0.0 and 1.0: " + str(x))
13 x = random.randrange(1, 100)
14 print("random number between 1 and 100: " + str(x))
```

```
math.sqrt(81) = 9.0
abs(-18) = 18
round(7.3) = 7
round(7.8) = 8
pi = 3.141592653589793
random number between 0.0 and 1.0: 0.04835491343133358
random number between 1 and 100: 89
```



```

1 # Program to display calendar of the given month and year
2
3 # importing calendar module
4 import calendar
5
6 # To take month and year input from the user
7 yy = int(input("Enter year: "))
8 mm = int(input("Enter month: "))
9
10 # display the calendar
11 print(calendar.month(yy, mm))

```

```

Enter year: 2020
Enter month: 9
      September 2020
Mo Tu We Th Fr Sa Su
      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

```



3 φίλοι αγόρασαν ένα λαχείο. Πρόγραμμα που διαβάζει τα κέρδη του λαχείου καθώς και το ποσό συμμετοχής του καθενός για την αγορά του. Να επιμερισθούν τα κέρδη αναλογικά με τα ποσά συμμετοχής.

```

1 # Epimerismos-kerdwn
2 k = float(input("Dwse kerdh: "))
3 p1 = float(input("Dwse poso symmetoxhs 1oy: "))
4 p2 = float(input("Dwse poso symmetoxhs 1oy: "))
5 p3 = float(input("Dwse poso symmetoxhs 1oy: "))
6 s = p1+p2+p3
7 x1 = p1/s*k
8 x2 = p2/s*k
9 x3 = p3/s*k
10 print("%.2f€ %.2f€ %.2f€ " %(x1, x2, x3))

```



```

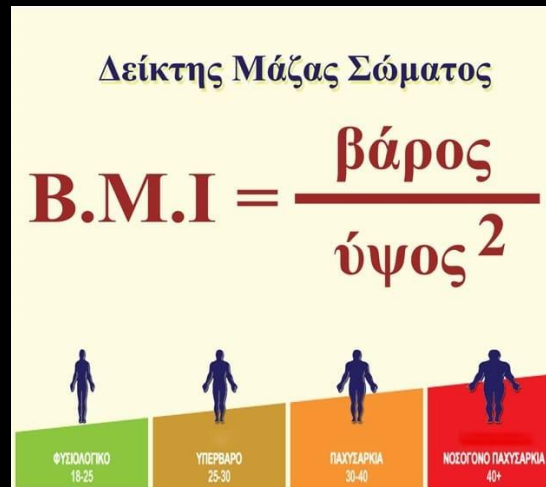
Dwse kerdh: 1000
Dwse poso symmetoxhs 1oy: 1
Dwse poso symmetoxhs 1oy: 2
Dwse poso symmetoxhs 1oy: 3
166.67€ 333.33€ 500.00€

```

```

1 baros = int(input("Dwse baros: "))
2 ypsos = float(input("Dwse ypsos: "))
3 baros = baros + 2
4 DMS = baros / pow(ypsos, 2)
5 ▸ if (DMS < 18.5):
6     answer = "Lipobarhs"
7 ▸ elif (DMS <= 24.9):
8     answer = "Kanonikos"
9 ▸ elif (DMS <= 29.9):
10    answer = "Yperbaros"
11 ▸ else:
12    answer = "Paxysarkos"
13
14 print(answer)

```



Dwse baros: 72  
Dwse ypsos: 1.75  
Kanonikos

#### Πρόγραμμα που

- δέχεται σαν είσοδο
  - το βάρος ενός ατόμου ( $\beta$ )
  - το ύψος του ( $u$ )
- υπολογίζει τον Δείκτη Μάζας Σώματος ( $\Delta M\Sigma$ ) ως εξής:  $\Delta M\Sigma = \beta / u^2$
- με βάση τον  $\Delta M\Sigma$  χαρακτηρίζει το άτομο ως:
  - Κάτω από 18.5 Λιποβαρής
  - 18.5 – 24.9 Κανονικός
  - 25.0 – 29.9 Υπέρβαρος
  - Πάνω από 30.0 Παχύσαρκος

$$x^2 + 4x + 1 = 0$$

$$\underline{x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

```

1 import math
2 a = float(input("a: "))
3 b = float(input("b: "))
4 c = float(input("c: "))
5 if (a == 0):
6     if (b == 0):
7         if (c == 0):
8             answer = "Aoristh"
9         else:
10            answer = "Adynath"
11    else:
12        p = -c/b
13        answer = "Mia riza = " + str(p)
14 else:
15     D = pow(b, 2) - 4 * a * c
16     if (D > 0):
17         p1 = (-b + math.sqrt(D)) / (2 * a)
18         p2 = (-b - math.sqrt(D)) / (2 * a)
19         answer = "Dyo rizes p1 = " + str(p1) + " p2 = " + str(p2)
20     elif (D == 0):
21         p = -b / (2 * a)
22         answer = "Mia diplh riza = " + str(p)
23     else:
24         answer = "Adynath"
25
26 print(answer)

```

a: 1

b: -5

c: 2

Dyo rizes p1 = 4.561552812808831 p2 = 0.438447187191169

a: 1

b: -2

c: 1

Mia diplh riza = 1.0

a: 1

b: 2

c: 3

Adynath



```
1 # Clock
2 import os
3 import time
4 import datetime
5
6 while (True):
7     today = datetime.datetime.now().strftime("%d/%m/%Y %H:%M:%S")
8     print("Current date & time (of python server):", today)
9     time.sleep(1)
10    os.system("clear")
```

```
|Current date & time (of python server): 28/09/2020 16:02:51
```



```

1 # Xelwnes
2 p1 = float(input("Dwse shmerino plithismo: "))
3 ep = float(input("Dwse % ethsia metabolh (<0): "))
4 p2 = int(input("Dwse plithismo - stoxo: "))
5 eth = 0
6 while (p1 > p2):
7     p1 = p1 + ep/100*p1
8     eth = eth + 1
9 print("Se %d xronia tha exoyn ginei %.0f" % (eth ,p1))

```

```

Dwse shmerino plithismo: 1000
Dwse % ethsia metabolh (<0): -3
Dwse plithismo - stoxo: 100
Se 76 xronia tha exoyn ginei 99

```

Άσκηση: προσαρμόστε το ώστε να λειτουργεί και για θετικό ρυθμό μεταβολής. Να βγάζει μήνυμα λάθους για μηδενικό ρυθμό.

```

Dwse shmerino plithismo: 1000
Dwse % ethsia metabolh: 3
Dwse plithismo - stoxo: 2000
Se 24 xronia tha exoyn ginei 2033

```

```

1 # xwres_plithismo2
2 on1 = input("onoma 1hs xwras:")
3 p1 = int(input("plithismnos 1hs xwras:"))
4 pos1 = float(input("% ethsio pososto metabolhs 1hs xwras:"))
5 on2 = input("onoma 2hs xwras:")
6 p2 = int(input("plithismnos 2hs xwras:"))
7 pos2 = float(input("% ethsio pososto metabolhs 2hs xwras:"))
8 if (p1<p2):
9     minp=p1
10    minpos=pos1
11    minon=on1
12    maxp=p2
13    maxpos=pos2
14    maxon=on2
15 else:
16    minp=p2
17    minpos=pos2
18    minon=on2
19    maxp=p1
20    maxpos=pos2
21    maxon=on1
22 if (minpos > maxpos):
23     eth=0
24     while (minp <= maxp):
25         minp=minp + minpos/100*minp
26         maxp=maxp + maxpos/100*maxp
27         eth=eth + 1
28         print("%d. %.0f %.0f" %(eth, minp, maxp))
29
30     print("Se %d xronia h xwra %s:%.0f kai h xwra %s:%.0f" %(eth, minon, minp, maxon, maxp))
31 else:
32     print("Adynato")

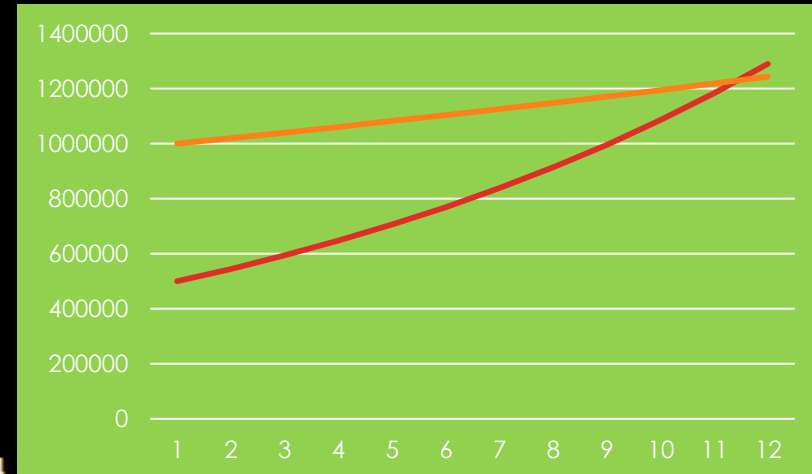
```

Να γραφεί πρόγραμμα που διαβάσει για 2 χώρες: τα ονόματα, τους σημερινούς πληθυσμούς και τους % ετήσιους ρυθμούς αύξησής τους. Να εμφανίζει σε πόσα χρόνια η μικρότερη σε πληθυσμό χώρα θα ξεπεράσει τη μεγαλύτερη ή το μήνυμα "δεν θα συμβεί ποτέ"

```

onoma 1hs xwras:x1
plithismnos 1hs xwras:500000
% ethsio pososto metabolhs 1hs xwras:9
onoma 2hs xwras:x2
plithismnos 2hs xwras:1000000
% ethsio pososto metabolhs 2hs xwras:2
1. 545000 1020000
2. 594050 1040400
3. 647514 1061208
4. 705791 1082432
5. 769312 1104081
6. 838550 1126162
7. 914020 1148686
8. 996281 1171659
9. 1085947 1195093
10. 1183682 1218994
11. 1290213 1243374
Se 11 xronia h xwra x1:1290213 kai h xwra x2:1243374

```



Άσκηση: υπολογίστε επιπλέον  
και τον αριθμό των ψηφίων

```
1 # Sum-of-digits
2 n=int(input("Enter a number:"))
3 tot=0
4 while(n>0):
5     dig=n%10
6     tot=tot+dig
7     n=n//10
8 print("The total sum of digits is:",tot)
```

Calculate the sum of the digits  
in an integer

5245

5 + 2 + 4 + 5

Sum of the digits

16

```
Enter a number:12345
The total sum of digits is: 15
```

**123456****Original Number****654321****Reversed Number**

```
1 # Reverse-number
2 n=int(input("Enter number: "))
3 rev=0
4 while(n>0):
5     dig=n%10
6     rev=rev*10+dig
7     n=n//10
8 print("Reverse of the number:",rev)
```

```
Enter number: 12345
Reverse of the number: 54321
```



## Divide and Conquer



```

1 import random
2
3 x = random.randrange(1,100)
4 tries = 0
5 found = False
6 while (not found):
7     tries = tries + 1
8     g = int(input("guess a number (1-100): "))
9     if (g < x):
10        print("upper")
11    elif (g > x):
12        print("lower")
13    else:
14        found = True
15
16 print("Found with " + str(tries) + " tries")

```

```

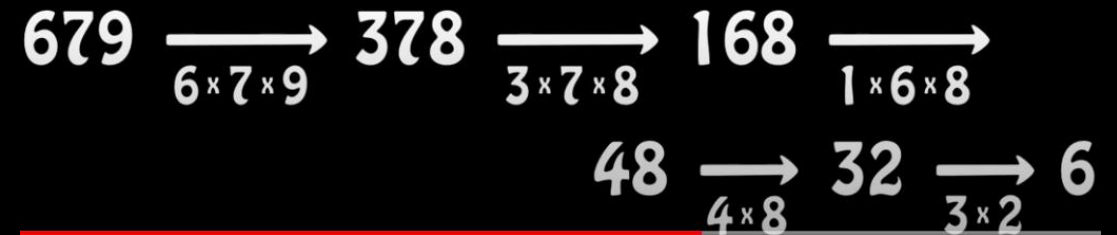
guess a number (1-100): 50
upper
guess a number (1-100): 75
lower
guess a number (1-100): 63
upper
guess a number (1-100): 69
lower
guess a number (1-100): 66
lower
guess a number (1-100): 65
Found with 6 tries

```

```

1 # pollaplasiastikh_epimonh
2 x = (int)(input("Dwse timh: "))
3 print("-----")
4 cnt = 0
5 while (x > 9):
6     g = 1
7     while (x != 0):
8         y = x % 10
9         g = g * y
10        x = x // 10
11    x = g
12    cnt = cnt + 1
13    print("%d. %d" %(cnt, x))
14
15 print("-----")
16 print("pollaplasiastikh_epimonh = %d" % cnt)

```



```

Dwse timh: 679
-----
1. 378
2. 168
3. 48
4. 32
5. 6
-----
pollaplasiastikh_epimonh = 5

```

Πολλαπλασιαστική επιμονή ενός ακέραιου είναι ο αριθμός των βημάτων που απαιτούνται ώστε το γινόμενο των ψηφίων του να είναι μονοψήφιος αριθμός. Π.χ. η πολλαπλασιαστική επιμονή του αριθμού 679 είναι 5, διότι:  $679 \rightarrow 6 \cdot 7 \cdot 9 = 378 \rightarrow 3 \cdot 7 \cdot 8 = 168 \rightarrow 1 \cdot 6 \cdot 8 = 48 \rightarrow 4 \cdot 8 = 32 \rightarrow 3 \cdot 2 = 6$ . 54. Να γραφεί πρόγραμμα το οποίο να διαβάζει έναν θετικό ακέραιο και να εμφανίζει την πολλαπλασιαστική επιμονή του καθώς και τα βήματα υπολογισμού της



```

1 # Repeated-menu
2 ep = 0
3 while (ep != 5):
4     print("1. Prosthesh")
5     print("2. Afairesh")
6     print("3. Pollaplasiasmos")
7     print("4. Diairesh")
8     print("5. Termatismos")
9     ep = int(input("Epilekse (1-5): "))
10    if (ep >= 1 and ep <= 4):
11        a = float(input("Dwse 1o oro: "))
12        b = float(input("Dwse 2o oro: "))
13        if (ep == 1):
14            print("-----> Result = %.2f" % float(a+b))
15        elif (ep == 2):
16            print("-----> Result = %.2f" % float(a-b))
17        elif (ep == 3):
18            print("-----> Result = %.2f" % float(a*b))
19        else:
20            print("-----> Result = %.2f" % float(a/b))
21    elif (ep != 5):
22        print("Lathos epilogh")
23
24 print("bye bye")

```

Προεκτάσεις:

1. ελέγξτε την περίπτωση διαίρεσης με το μηδέν(0)
2. προσθέστε και άλλες πράξεις (τετραγωνική ρίζα κλπ.)

```

1. Prosthesh
2. Afairesh
3. Pollaplasiasmos
4. Diairesh
5. Termatismos
Epilekse (1-5): 3
Dwse 1o oro: 6
Dwse 2o oro: 8
-----> Result = 48.00
1. Prosthesh
2. Afairesh
3. Pollaplasiasmos
4. Diairesh
5. Termatismos
Epilekse (1-5): 4
Dwse 1o oro: 7
Dwse 2o oro: 2
-----> Result = 3.50
1. Prosthesh
2. Afairesh
3. Pollaplasiasmos
4. Diairesh
5. Termatismos
Epilekse (1-5): 5
bye bye

```

```

1 # Propaideia
2 for x1 in range(1, 11):
3     for x2 in range(1, 11):
4         print(str(x1) + " x " + str(x2) + " = " + str(x1*x2))

```

Άσκηση: triangle-with-numbers

```

1
12
123
1234
12345
123456
1234567
12345678
123456789
12345678
1234567
123456
12345
1234
123
12
1

```

```

1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
1 x 4 = 4
1 x 5 = 5
1 x 6 = 6
1 x 7 = 7
1 x 8 = 8
1 x 9 = 9
1 x 10 = 10
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20

```

```

8 x 10 = 80
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
10 x 10 = 100

```

```
1 #Christmas-Tree (Python 3)
2 N = 20
3 # fylla
4 for x in range(1, N+1):
5     for y in range(1, (int)((N-x)/2)):
6         print(" ", end = ' ') # no new line
7     for y in range(1, x+1):
8         print("*", end = ' ')
9     print("")
10
11 # kormos
12 for x in range(1, (int)(N/3)):
13     for y in range(1, (int)(N/2.5)):
14         print(" ", end = ' ')
15     for y in range(1, (int)(N/3)):
16         print("*", end = ' ')
17     print("")
```

```
          *
          **
          ***
          ****
          *****
          ******
          *******
          ********
          *********
          **********
          **********
          **********
          **********
          **********
          **********
          **********
          *****
          *****
          *****
          *****
          *****
```

```

1 #Binary System
2
3 for x1 in range(0, 2):
4     for x2 in range(0, 2):
5         for x3 in range(0, 2):
6             for x4 in range(0, 2):
7                 for x5 in range(0, 2):
8                     for x6 in range(0, 2):
9                         for x7 in range(0, 2):
10                            for x8 in range(0, 2):
11                               print(x1, x2, x3, x4, x5, x6, x7, x8)

```

```

100110001000101000
101000010100101001
001101001000000101
101001001001001010
001001010101010110

```

```

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1
0 0 0 0 0 0 1 0
0 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0
0 0 0 0 0 1 0 1
0 0 0 0 0 1 1 0
0 0 0 0 0 1 1 1
0 0 0 0 1 0 0 0
0 0 0 0 1 0 0 1
0 0 0 0 1 0 1 0
0 0 0 0 1 0 1 1
0 0 0 0 1 1 0 0
0 0 0 0 1 1 0 1
0 0 0 0 1 1 1 0
0 0 0 0 1 1 1 1
0 0 0 1 0 0 0 0
0 0 0 1 0 0 0 1
0 0 0 1 0 0 1 0
0 0 0 1 0 0 1 1
0 0 0 1 0 1 0 0

```

```

1 1 1 0 1 1 0 1
1 1 1 0 1 1 1 0
1 1 1 0 1 1 1 1
1 1 1 1 0 0 0 0
1 1 1 1 0 0 0 1
1 1 1 1 0 0 1 0
1 1 1 1 0 0 1 1
1 1 1 1 0 1 0 0
1 1 1 1 0 1 0 1
1 1 1 1 0 1 1 0
1 1 1 1 0 1 1 1
1 1 1 1 1 0 0 0
1 1 1 1 1 0 0 1
1 1 1 1 1 0 1 0
1 1 1 1 1 0 1 1
1 1 1 1 1 1 0 0
1 1 1 1 1 1 0 1
1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1

```

...

Άσκηση: συνδυασμοί 2 ζαριών



|     |     |
|-----|-----|
| 1 1 | 5 1 |
| 1 2 | 5 2 |
| 1 3 | 5 3 |
| 1 4 | 5 4 |
| 1 5 | 5 5 |
| 1 6 | ... |
| 2 1 | 6 1 |
| 2 2 | 6 2 |
| 2 3 | 6 3 |
| 2 4 | 6 4 |
| 2 5 | 6 5 |
| 2 6 | 6 6 |

```

1 #Letter-By-Letter
2 Sentence = input("Input a sentence: ")
3 size = len(Sentence)
4 for i in range(0, size):
5     for j in range(0, i+1):
6         print(Sentence[j], end = '')
7     print("")

```

### Άσκηση: αντιστροφή κειμένου

ESREVER

Input a sentence: ΤΡΙΤΟ ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙΟΥ

Inverse sentence: ΥΟΙΕΛΚΑΡΗ ΟΙΕΚΥΛ ΟΚΙΝΕΓ ΟΤΙΡΤ

Input a sentence: ΤΡΙΤΟ ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙΟΥ

T

TP

TPI

TPIT

TPITO

TPITO

TPITO Γ

TPITO ΓΕ

TPITO ΓΕΝ

TPITO ΓΕΝΙ

TPITO ΓΕΝΙΚ

TPITO ΓΕΝΙΚΟ

TPITO ΓΕΝΙΚΟ

TPITO ΓΕΝΙΚΟ Λ

TPITO ΓΕΝΙΚΟ ΛΥ

TPITO ΓΕΝΙΚΟ ΛΥΚ

TPITO ΓΕΝΙΚΟ ΛΥΚΕ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ Η

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙΟ

TPITO ΓΕΝΙΚΟ ΛΥΚΕΙΟ ΗΡΑΚΛΕΙΟΥ



```
1 # Trapeza
2 k = float(input("Dwse kefalαιο katatheshs: "))
3 ep = float(input("Dwse % epitokio: "))
4 x = int(input("Dwse xronia katatheshs: "))
5 for i in range(1, x+1):
6     k = k + ep/100*k
7 print("%.2f €" % k)
```

```
Dwse kefalαιο katatheshs: 500
Dwse % epitokio: 1.2
Dwse xronia katatheshs: 10
563.35 €
```



```

1 #Prwtos-Arithmos
2 x = int(input("Dwse akeraio: "))
3 diairetes = 0
4 for i in range(1, x+1):
5     if (x % i == 0):
6         diairetes = diairetes + 1
7
8 if (diairetes == 2):
9     answer = "Prwtos"
10 else:
11     answer = "Mh prwtos. Exei " + str(diairetes) + " diairetes"
12 print(answer)

```

## PRIME NUMBERS

2, 3, 5, 7, 11, 13, 17, 19, 23,  
29, 31, 37, 41, 43, 47, 53, 59,  
61, 67, 71, 73, 79, 83, 89, 97

```

Dwse akeraio: 17
Prwtos

```

```

Dwse akeraio: 12
Mh prwtos. Exei 6 diairetes

```

## PRIME NUMBERS

2, 3, 5, 7, 11, 13, 17, 19, 23,  
29, 31, 37, 41, 43, 47, 53, 59,  
61, 67, 71, 73, 79, 83, 89, 97

```

1 #Prwtoi-Arithmoi
2 a = int(input("Dwse arxikh timh: "))
3 b = int(input("Dwse telikh timh: "))
4 p = 0;
5 for x in range(a, b+1):
6     diairetes = 0
7     for i in range(1, x+1):
8         if (x % i == 0):
9             diairetes = diairetes + 1
10
11     if (diairetes == 2):
12         answer = str(x) + ": Prwtos"
13         p = p + 1
14     else:
15         answer = str(x) + ": Mh prwtos. Exei " + str(diairetes) + " diairetes"
16     print(answer)
17
18 print ("Apo to " + str(a) + " mexri to " + str(b) + " brethikan " + str(p) + " prwtoi")

```

```

Dwse arxikh timh: 1
Dwse telikh timh: 20
1: Mh prwtos. Exei 1 diairetes
2: Prwtos
3: Prwtos
4: Mh prwtos. Exei 3 diairetes
5: Prwtos
6: Mh prwtos. Exei 4 diairetes
7: Prwtos
8: Mh prwtos. Exei 4 diairetes
9: Mh prwtos. Exei 3 diairetes
10: Mh prwtos. Exei 4 diairetes
11: Prwtos
12: Mh prwtos. Exei 6 diairetes
13: Prwtos
14: Mh prwtos. Exei 4 diairetes
15: Mh prwtos. Exei 4 diairetes
16: Mh prwtos. Exei 5 diairetes
17: Prwtos
18: Mh prwtos. Exei 6 diairetes
19: Prwtos
20: Mh prwtos. Exei 6 diairetes
Apo to 1 mexri to 20 brethikan 8 prwtoi

```



```

1 #Minimum-Coins
2 poso = int(input("Dwse akeraio poso (€): "))
3 Coins = [500, 200, 100, 50, 20, 10, 5, 2, 1]
4 i = 0
5 while (poso != 0):
6     x = int(poso/Coins[i])
7     if (x != 0):
8         print(str(x) + " coins of " + str(Coins[i]) + " €")
9         poso = poso % Coins[i]
10    i = i + 1

```

```

Dwse akeraio poso (€): 123456
246 coins of 500 €
2 coins of 200 €
1 coins of 50 €
1 coins of 5 €
1 coins of 1 €

```

```
1 # Lotto_lucky_numbers_generator
2
3 import random
4 Numbers = [0, 0, 0, 0, 0, 0]
5 count = 0
6 while (count < 6):
7     x = random.randrange(1, 49)
8     # search x in Numbers
9     found = False
10    for i in range(0, 6):
11        if (Numbers[i] == x):
12            found = True
13            break
14    if (found == False): # 1st time met number
15        count = count + 1
16        Numbers[count-1] = x;
17
18 Numbers.sort()
19 for i in range(0, 6):
20     print(str(Numbers[i]) + " ", end = '')
21
22 print("")
```



6 12 32 34 37 46