

TasPython

1η Σειρά Ασκήσεων

Λύσεις

TasPython Team
www.taspython.tk

3 Απριλίου 2009

1 Δομές ελέγχου

1.1 Hello World

Εγκαταστήστε την *Python 3k* και γράψτε ένα πρόγραμμα που να τυπώνει στην οθόνη σας το Hello World.

Προαιρετικά: Γράψτε έναν οδηγό για το πως κάνατε την εγκατάσταση και σε ποιο σύστημα.

Λύση:

```
#!/usr/bin/python
# Filename: hello.py
print('Hello World')
```

Author: BadGuy

1.2 Απλό menu εστιατορίου

Γράψτε ένα απλό μενού εστιατορίου, όπου παρουσιάζεται ένα μενού στον πελάτη, όπως για παράδειγμα :

1. Κολοκυθάκια γιαχνί
2. Πατάτες στα κάρβουνα
3. Κοτόπουλο γεμιστό

4. Σαλιγκάρια

Τι θα επιθυμούσατε;

Και ανάλογα με την επιλογή που δίνει ο χρήστης να εμφανίζεται ένα κατάλληλο μήνυμα, ή μήνυμα λάθος αν δεν ανήκει στο εύρος αποδεκτών τιμών.

Λύση:

```
#!/usr/bin/python
# Filename: menu.h

sum = 0

print('Welcome to our restaurant!!!\nPlease select your meal by giving one of the numbers below')
print('1.Kolokythakia giaxni\n2.Patates sta karvouna\n3.Kotopoulo gemisto\n4.Saligaria\n5.exit')

while True:
    preference = int(input('Eisagetai tin epilogi sas : '))

    if preference == 1:
        print('Freska kolokythakia erxontai!!')
        print('Gia synexeia tis paraggelias patiste 1-4, alliws 5 gia exodo')
        sum += 5
    elif preference == 2:
        print('Laxtaristes patates erxontai!!')
        print('Gia synexeia tis paraggelias patiste 1-4, alliws 5 gia exodo')
        sum += 4
    elif preference == 3:
        print('Kotopoulo alaniariko erxetai!!')
        print('Gia synexeia tis paraggelias patiste 1-4, alliws 5 gia exodo')
        sum += 8
    elif preference == 4:
        print('Saligaria mpoumpourista erxontai!!')
        print('Gia synexeia tis paraggelias patiste 1-4, alliws 5 gia exodo')
        sum += 10
    elif preference == 5:
        print('Euxaristoume gia tin pritimisi sas!!Euxomaste na sas ksanadoume syntoma!')
        break
    else:
        print('Lathos epilogi, prospathiste pali!!!')

print('H paraggelia sas kostizei {0} eurw, kali sas oreksi!!!'.format(sum))
```

Author: BadGuy

Την διάλεξα γιατί έχει αυτό με το λογαριασμό.

```
#!/usr/bin/python3

while True:
    print('-----')
    print('Dialekse apo to menu mas')
    print('1. Mprizola')
    print('2. Kontosoubli')
    print('3. Patates')
```

```

print('4. menu tou Myle')
print('exodos->oti leei den tha sas eksigw ta panta')
print('-----\n\n')

fai=input('Ti tha fas??? ')
if fai=='1':
    print('Kreeeeeeeeas\n')
elif fai=='2':
    print('Perissotero kreas!\n')
elif fai=='3':
    print('Would you like fries with that?\n')
elif fai=='4':
    print('Oute ston exthro mou\n')
elif fai=='exodos':
    break
else :
    print('Re mas douleyeis\n')

```

Author: gsfak

Την διάλεξα γιατί είναι το μόνο που δεν κρασάρει για οποιαδήποτε είσοδο.

1.3 Εύρεση ριζών 2ου Βαθμού Εξίσωσης

Υπολογίστε τις λύσεις μια εξίσωσης δευτέρου βαθμού, ζητώντας από τον χρήστη τους συντελεστές α, β, γ .

Bonus: Υπολογίστε και μιγαδικές ρίζες αν η διακρίνουσα είναι αρνητική.

Λύση:

```

from math import sqrt

a = int(input("Please insert the coefficient of the x^2: "))
b = int(input("Please insert the coefficient of the x: "))
c = int(input("Please insert the constant: "))

if a == 0:
    print("This is not a 2nd degree equation")
    import sys
    exit()

D = b**2 - 4 * a * c

if D >= 0:
    x1 = (-b + sqrt(D)) / (2 * a)
    x2 = (-b - sqrt(D)) / (2 * a)
else:
    x1 = complex(-b / (2 * a), sqrt(-D) / 2 * a)
    x2 = complex(-b / (2 * a), -sqrt(-D) / 2 * a)

sf = "x1 = {0}, x2 = {1}"
print(sf.format(x1, x2))

```

Author: myle

Όλοι εκτός από τον gsfak έκαναν παρόμοια λύση.

2 Βρόγχοι

2.1 Αριθμοί Fibonacci

Βρείτε τους 20 πρώτους αριθμούς Fibonacci.

Δίνεται ότι:

$$F_n = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F_{n-1} + F_{n-2}, & \text{else} \end{cases} \quad (1)$$

Λύση:

```
# That was taken from the official Python 3k tutorial because is the best way
# to find the fibonacci numbers...

# the 2 first fibonacci numbers are always known
a, b = 0, 1

# print the first fibonacci number
print(a, end = ' ')

# find the 20 first fibonacci numbers and print them
for i in range(1, 20):
    print(b, end = ' ')
    a, b = b, a + b

print("\n")
```

Author: sbos-x

Είναι η καλύτερη λύση. Και καλό είναι να ρίξετε μια ματιά στο πως κάνει την ανάθεση. Άλλες γλώσσες δεν το επιτρέπουν.

2.2 Άθροιση πολλών αριθμών

Προσθέστε όλους του φυσικούς αριθμούς που είναι κάτω του 1000 και είναι πολλαπλάσια του 3 και του 7.

Λύση:

```
sum(i for i in range(1, 1001) if not (i%3 == 0 or i%7 ==0))
```

Author: myle

Αυτό μπήκε εδώ γιατί μου αρέσουν τα μονογραμμικά δημιουργήματα. Απλά θέλει ένα print.

```
#!/usr/bin/python
# Filename: sum.py

sum = 0

for i in range(0,1000):
    if (i%3 == 0) or (i%7 == 0):
        sum += i

print('The sum is', sum)
```

Author: BadGuy

Αυτό έχει print.

3 Χειρισμός Αλφαριθμητικών

3.1 Αντιστροφή Αλφαριθμητικού

Δημιουργείστε ένα πρόγραμμα που δέχεται ως είσοδο μια πρόταση και την αντιστρέφει.

Λύση:

```
#!/usr/bin/python
# Filename: reverse.py

def reverse(text):
    return text[::-1]

sentence = input('Enter your text: ')
invert = reverse(sentence)

print('The inverted text is', invert)
```

Author: BadGuy

Όλα γίνονται πιο όμορφα με μία συνάρτηση.

3.2 Τεμαχισμός Αλφαριθμητικού

Δημιουργείστε ένα πρόγραμμα που κόβει ένα αλφαριθμητικό σε συγκεκριμένους χαρακτήρες

Λύση:

```
#!/usr/bin/python

stg=input('Dwse ena string ')
spl=input('Dwse xarakthra alliws den exei split ')

while len(spl)!=1:
    spl=input('Dwse xarakthra eipame... ')

list1=[]
i=0;

for j in range(len(stg)):

    if stg[j]==spl:

        list1.append(stg[i:j])
        i=j+1

list1.append(stg[i:len(stg)])
print(list1)
```

Author: gsfak

Δεν υπήρχε κανένας λόγος επιλογής ήταν παρόμοιο με του βαδ.

3.3 Στατιστικά Εγγράφου

Δημιουργείστε ένα πρόγραμμα που :

1. Μειράει πόσοι χαρακτήρες υπάρχουν σε ένα αρχείο
2. Πόσες λέξεις υπάρχουν
3. Πόσες προτάσεις υπάρχουν
4. Πόσοι χαρακτήρες νέας γραμμής

Λύση:

```
# count lines, sentences, and words of a text file
# set all the counters to zero
```

```

lines, blanklines, sentences, words = 0, 0, 0, 0

try:
    # use a text file you have, or google for this one ...
    filename = 'random_file'
    textf = open(filename, 'r')
except IOError:
    print('Cannot open file {1} for reading', filename)
    import sys
    sys.exit(0)

# reads one line at a time
for line in textf:
    print(line) # test
    lines += 1

    if line.startswith('\n'):
        blanklines += 1
    else:
        # assume that each sentence ends with . or ! or ?
        # so simply count these characters
        sentences += line.count('.') + line.count('!') + line.count('?')

        # create a list of words
        # use None to split at any whitespace regardless of length
        # so for instance double space counts as one space
        tempwords = line.split(None)
        print(tempwords) # test

        # word total count
        words += len(tempwords)

textf.close()

print("Lines      : ", lines)
print("Blank lines: ", blanklines)
print("Sentences  : ", sentences)
print("Words      : ", words)

```

Author: myle

Δείτε πως είναι ένα try block.

4 Συνδυαστικές Ασκήσεις

4.1 Εύρεση πρώτων αριθμών

Βρείτε όλους τους πρώτους αριθμούς που βρίσκονται ανάμεσα στο 1 έως το 100.

Υπόδειξη: Μπορείτε να ψάχνετε αν ένας αριθμός έχεις διαιρέτες μέχρι τον προς τα κάτω ακέραιο της τετραγωνικής του ρίζας.

4.1.1 Αλγόριθμος του Ερατοσθένη

1. Δημιουργείστε μια πλήρη λίστα με όλους τους αριθμούς $[2, n]$.

2. Αφαιρέστε τα πολλαπλάσια του 2
3. Ο επόμενος αριθμός στην λίστα που έχει μείνει είναι πρώτος
4. Αφαιρέστε όλα τα πολλαπλάσια του
5. Επαναλάβετε τα βήματα 3 – 4 μέχρι να φθάσετε έναν αριθμό που είναι μεγαλύτερος από $\lceil \sqrt{n} \rceil$.
6. Οι αριθμοί που απομένουν είναι πρώτοι

Λύση:

```
from math import sqrt

# the desirable number of numbers
n = 100

# all the numbers in the desirable range
x = [i for i in range (2, 101)]
# filter out all the even numbers
x = [i for i in x if not (i % 2 ==0 and i != 2)]

counter = 0
prime = x[counter]

while prime < sqrt(n) + 1:
    # the first number in the list is prime
    prime = x[counter]
    print(prime)
    print(x)
    # filter out all its multiples
    x = [i for i in x if not (i % prime ==0 and i != prime)]
    counter = counter + 1
    print('a')

print(x)
```

Author: myle

Είναι πολύ ενδιαφέρουσα η δυνατότητα της πηπτηον να κάνει ανάθεση σε λίστες με αυτόν το τρόπο που φαίνεται εδώ. Συγκεκριμένα είναι ο αλγόριθμος του Ερατοσθένη.

Προτείνω να ρίξετε μια ματιά σε όλες τις ασκήσεις. Έχουν όλες διαφορετικό στυλ προγραμματισμού. Μερικές έχουν λάθη, αλλά όχι τίποτα ιδιαίτερο. Ελπίζω να μην μου ξέφυγε κάτι.