

Laboratorio di Python

Alcuni moduli utili, Esercizi vari

Sara Zuppiroli

Università di Bologna

10 e 12 aprile 2013



Sommario

- 1 Correzione esercizi
- 2 Alcuni Moduli Utili
- 3 Scegliere una struttura dati
- 4 Esercizi per casa



Esercizi

Scrivere e documentare le funzioni che risolvano i seguenti problemi:

- 1 Definire una funzione che presa una sequenza come parametro restituisca il valore della media geometrica di tale sequenza
- 2 Definire una funzione che presa una terna di valori come parametro mi dica se questa é una terna pitagorica o meno
- 3 Definire una funzione che presa un sequenza mi restituisca tutti i possibili suffissi di tale sequenza (es. $(1,4,3) \rightarrow (), (3,), (4,3), (1,4,3))$ usare l'iterazione e la ricorsione
- 4 Definire una funzione che presa un sequenza e un parametro intero mi restituisca tutti i possibili suffissi di tale sequenza fino alla lunghezza definita dal parametro intero (es. preso $(1,4,3), 1 \rightarrow (), (3,))$ usare l'iterazione e la ricorsione



Terna pitagorica

```
def geom_media(l):  
    #Funzione che calcola la media geometrica  
    p = 1      #Inizializzo la variabile per il prodotto degli elementi  
    for e in l:  
        p*=e  
    return p**(1/len(l))
```

Cosa si poteva migliorare? Se richiamo la funzione col valore 4



Terna pitagorica

```
def terna_pitagorica():  
    l1=eval(input('inserire un cateto'))  
    l2=eval(input('inserire un cateto'))  
    l3=eval(input('inserire l'ipotenusa'))  
    if l3==sqrt(l1**2+l2**2):    #se vero allora terna pitagorica  
        return True  
    return False
```

Cosa é sbagliato?



Terna pitagorica

```
def terna_pitagorica(l):  
    #Funzione che stabilisce se terna e pitagorica  
    if len(l)==3:  
        l.sort()           #Ordino la lista in ordine crescente  
        return l[0]**2+l[1]**2==l[2]**2  
    else:  
        print('La funzione non e stata chiamata su una terna')
```



Suffissi

```
def suffix(l):  
    #Funzione che restituisce i possibili suffissi di una lista  
    r = []  
    for i in range(len(l)+1):  
        r.append(l[i:])  
    return r
```



Suffissi 2

```
def suffix_lim(l,n):  
    #Funzione che restituisce i possibili suffissi di una  
    #lista di lunghezza fino a n  
    r = []  
    for i in range(len(l),len(l)-n-1,-1):  
        r.append(l[i:])  
    return r
```



Errori

- 1 Qualcuno ha definito un menú che non era stato richiesto
- 2 Qualcuno non ha saputo fare tutti gli esercizi
- 3 Qualcuno non ha preso come parametri delle funzioni i dati richiesti



Modulo random

- 1 `import random`
- 2 `random.random()` (numero casuale tra 0.0 e 1.0)
- 3 `random.uniform(a, b)` (numero casuale tra a e b)
- 4 `random.randint(n, m)` (intero casuale tra n e m, estremi inclusi)
- 5 `random.randrange(n, m, k)` (intero casuale in range(n,m,k))
- 6 `random.choice(s)` (sceglie a caso un elemento da una successione s)
- 7 `random.sample(s, n)` (estrazione casuale senza reinserimento di n elementi da una successione s)

Maggiori informazioni:

<http://docs.python.org/library/random.html>.



Manipolazione stringhe

- Il tipo `str` ha dei metodi interessanti come `split` e `splitlines`.

```
❶ line='Il mio cane si chiama Fido.'
❷ line.split() [= ['Il', 'mio', 'cane', 'si',  
  'chiama', 'Fido.']]
❸ text='Mary had a little lamb, Its fleece was  
  white as snow;  
  And everywhere that Mary went,  
  The lamb was sure to go.'
❹ text.splitlines() [= ['Mary had a little lamb,',  
  'Its fleece was white as snow;', 'And everywhere  
  that Mary went,', 'The lamb was sure to go.']]
```

- Vedi anche <http://docs.python.org/library/stdtypes.html#string-methods>.



Leggere file

Python ha un tipo di dato File, completo di una serie di metodi.

- ❶ `f = open('./esempio.py', 'r')`
apre il file `'./esempio.py'` in lettura (`r=read`);
- ❷ `lines = f.readlines()`
contiene la lista di tutte le righe del file.
- `f.close()` richiamo il metodo `close` sull'oggetto `f`. È importante chiudere il file una volta aperto.
- Per i più curiosi, altre informazioni su <http://docs.python.org/library/stdtypes.html#file-objects>.



Scrivere su file

```
fo=file('prova.txt', 'w')
fo.write('Primo file di testo:\n\n')
fo.write('terza riga,\n')
fo.write('quarta riga riga,\n')
fo.write('fine del file.')
fo.close()
```

- 1 `fo = open('prova.txt', 'w')` apro il file prova in scrittura.
fo ora é un'istanza della classe file. (fo oggetto)
- 2 `fo.write('Primo file di testo:
n
n')` richiamo il metodo write col testo che desidero scrivere
(Primo file di testo:
n
n) della classe file al mio oggetto fo.
- 3 `fo.close()` richiamo il metodo close sull'oggetto fo.



Esercizio

- 1 Scrivere una funzione che dato il nome di un file scelga casualmente una parola presente in tale file.



Scompongo l'esercizio in sotto-problemi

- 1 Scrivere una funzione che apra un file e restituisca tale file come una lista.
- 2 Scrivere una funzione che presa una stringa elimini alcuni caratteri speciali e restituisca la stringa pulita.
- 3 Scrivere una funzione che data una stringa definisca se questa é una parola (si definisce parola una concatenazione di sole lettere.)
- 4 Scrivere una funzione che data una lista di stringhe restituisca la lista delle sole parole distinte presenti tra le stringhe.
- 5 Scrivere una funzione che dato un file restituisca casualmente una parola appartenente al file



Apri il file; Elimina punteggiatura

```
import random
import string
def openfile(nome):
    f=open(nome,'r')
    r=f.readlines()
    f.close()
    return r
```

```
def eliminapunteggiatura(s):
    #elimino la punteggiatura nella parola
    s=s.lstrip("`'.,;:!?"+'""'}''-').rstrip("`'.,;:!?"+'""'}'')
    return s
```



Riconosco se é parola

```
def def_parola(s):  
    def_parola=True  
    for i in range(len(s)):  
        if not(s[i] in string.letters):  
            def_parola=False  
            break  
    else:  
        def_parola=True  
    return def_parola
```



Dalla lista di stringhe alla lista di parole

```
def suddivido_parole(r):  
    #data una lista di stringhe prende solo le parole  
    w=[] #variabile per la lista delle parole  
    for line in r: #per ogni riga del file  
        for s in line.split(): #per ogni stringa  
            s=eliminapunteggiatura(s)  
            parola=def_parola(s) #verifico se e parola  
            if (s not in w) and parola:  
                w.append(s)  
  
    return w
```



Estrazione casuale di una parola da un file

```
def random_word_from_file(nome):  
    return random.choice(suddivido_parole(openfile(nome)))
```



Esercizio

Scrivere una funzione che stampi la lista delle linee contenute nel file dove é salvata la funzione stessa. Cos'é un programma? Sapreste spiegare come funziona questa funzione?



Matrici diagonali

Quale può essere una buona struttura dati per la memorizzazione e l'implementazione delle operazioni con le matrici diagonali? Una matrice diagonale é una matrice quadrata in cui solamente i valori della diagonale principale possono essere diversi da 0.



Soluzione

1 Matrice = [[1,0,0], [0,2,0], [0,0,4]]

2 Matrice = [1,2,4]

Perché scegliere la seconda rappresentazione?



Esercizio

Definiamo le funzioni che implementino le operazioni di:

- 1 somma
- 2 moltiplicazione

tra matrici diagonali. Definire le funzioni che data una matrice diagonale ne calcoli il:

- 1 determinante
- 2 la matrice inversa



Esercizio

Definiamo le funzioni che implementino le operazioni di:

- 1 somma
- 2 moltiplicazione

indipendentemente dalla rappresentazione della matrice che é utilizzata.



Esercizi

Scrivere e documentare le funzioni che risolvano i seguenti problemi:

- 1 Definire una funzione che presa una sequenza come parametro restituisca il numero dei valori appartenenti a tale sequenza e che soddisfano una determinata caratteristica. Ad esempio una caratteristica potrebbe essere la definizione di elemento *valle*: un elemento appartenente a una determinata sequenza si definisce valle se $s[i] < s[i-1]$ e $s[i] < s[i+1]$. Data una sequenza quanti elementi valle sono presenti in tale sequenza?
- 2 Scrivere una funzione che presi due file copi il primo nel secondo file.
- 3 Scrivere una funzione che simuli l'estrazione del lotto, tale funzione deve scrivere su un file il risultato dell'estrazione.

Inviare gli esercizi svolti a: **labinfo.mat.unibo@gmail.com**



Cosa abbiamo fatto?

- 1 Correzione esercizi
- 2 Alcuni Moduli Utili
- 3 Scegliere una struttura dati
- 4 Esercizi per casa

