

Laboratorio di Python

Ricorsione

19 marzo 2015

Sommario

- 1 Correzione esercizi
- 2 Esercizi per casa
- 3 Ricorsione
- 4 Comprendere il codice

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 prese due sequenze rappresentanti gli elementi di un insieme come parametro ne restituisca le operazioni di unione, intersezione, differenza e prodotto cartesiano.
- 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
- 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



Correzione

```
def media_geo(c):  
    somma=0  
    for i in range(0,len(c)):  
        somma=somma+c[i]**2  
    return math.sqrt(somma)  
  
def operazioni_insiemi(a,b):  
    unione=b  
    differenza=()  
    intersezione=()  
    cartesiano=()  
    for i in range(0,len(a)):  
        if a[i] in b and a[i] not in intersezione:  
            intersezione=intersezione+a[i]  
        if a[i] not in b and a[i] not in differenza:  
            differenza=differenza+a[i]  
    if a[i] not in unione:  
        unione= unione+a[i]  
    for i in range(0,len(a)):  
        for j in range(0,len(b)):  
            cartesiano=cartesiano+((a[i],b[j]),)  
    return(unione,differenza,intersezione,cartesiano)
```



Correzione

```
def suffissi(a):  
    c=()  
    t=(),)  
    for i in range(0, len(a)):  
        c=c+a[i]  
        t=t+(c,)  
    return t  
def suffissi_limitato(a,l):  
    c=()  
    t=(),)  
    for i in range(0, l):  
        c=c+a[i]  
        t=t+(c,)  
    return t
```



Funzione ricorsiva

```
def palin(s):  
    x=len(s)  
    if x==0 or x==1:  
        return 'palindroma'  
    if s[0]==s[x-1]:  
        return palin(s[1:x-1])  
    else:  
        return 'non palindroma'
```

- Cerchiamo di capire come si esegue questa funzione.



Esempio di esecuzione

- `palin('ansna')`
- $x \rightarrow 5$
- $5 == 0$ or $5 == 1 \rightarrow false$
- $a == a \rightarrow true \rightarrow palin('nsn')$
- $x \rightarrow 3$
- $3 == 0$ or $3 == 1 \rightarrow false$
- $n == n \rightarrow true \rightarrow palin('s')$
- $x \rightarrow 1$
- $1 == 0$ or $1 == 1 \rightarrow true \rightarrow palindroma$



Trasformare l'iterazione in ricorsione

```
def fun(s):  
    r=''  
    i=0  
    while (i <len(s)):  
        r=r+s[len(s)-1-i]  
        i=i+1  
    return r
```

- Trasforma questa funzione in una funzione ricorsiva.



Ricorsione

Cosa computa questa funzione?

```
def fun(s, r):  
    if s=='':  
        return r  
    else:  
        r=r+s[len(s)-1]  
        return fun(s[0:len(s)-1], r)
```



Esercizio

Un maestro, per mettere a tacere gli allievi, disse loro di fare la somma di tutti i numeri da 1 a n . Un bambino fece tale somma dopo pochi minuti, grazie a quale soluzione?

Codice

```
def f_ite(n):  
    if n<=0:  
        return ('errore')  
    else:  
        s=0  
        for i in range(1,n+1):  
            s=s+i  
        return s  
  
def f_ric(n):  
    if n<=0:  
        return('errore')  
    if n==1:  
        return 1  
    else:  
        return(n+f_ric(n-1))  
  
def f(n):  
    if n<=0:  
        return ('errore')  
    else:  
        return(n*(n+1)//2)
```



Testiamo le soluzioni

- Proviamo le nostre funzioni con diversi input
- 3,10,100
- vediamo qualche differenza percettibile nell'esecuzione dei nostri programmi?

Esercizi

- Scrivere una funzione ricorsiva che calcoli la somma dei primi n numeri dispari.
- Scrivere una funzione ricorsiva che prenda presi due numeri c , p restituisca c^p
- Scrivere una funzione ricorsiva che prenda presi due numeri c , p restituisca $c * p$



La doppia ricorsione

Data la seguente funzione così definita:

$$F(0) = 1$$

$$F(n) = F_{n-1} + F_{n-1}$$

Scrivere una funzione che prenda come parametro n , e restituisca $F(n)$. Quanto valgono $F(2)$ ed $F(3)$? Cosa calcola questa funzione?



Cosa stampa questo codice?

```
def r(s):  
    c=0  
    for x in s:  
        if x>0:  
            c+=x  
        else:  
            x=-x  
    return c
```

```
L=(2,5,-3,-2)  
print(r(L))  
print(L)
```

Cosa stampa questo codice?

```
def r(s):  
    c=0  
    for x in s:  
        if x>0:  
            c+=x  
        else:  
            x=-x  
    return c
```

```
L=(2,5,-3,-2)
```

```
print(r(L))
```

```
print(L)
```

```
F=[1,-3,2,-1]
```

```
print(r(F))
```

```
print(F)
```



Esercizi a casa

- Scrivere un unico programma, che attraverso un menù di selezione restituisca i risultati delle funzioni che seguono, senza uscire dal programma se non selezionando il numero 4.
 - la somma dei primi n numeri pari, dove n è richiesto in input (soluzione iterativa con for)
 - la somma dei primi n numeri dispari, dove n è richiesto in input (soluzione iterativa con while)
 - la serie geometrica di n numeri, dove n è richiesto in input (soluzione ricorsiva)
- Inviare gli esercizi svolti a: **labinfo.mat.unibo@gmail.com**



Sommario

- 1 Correzione esercizi
- 2 Esercizi per casa
- 3 Ricorsione
- 4 Comprendere il codice