

# Laboratorio di Python

## Iterazione indeterminata

17 marzo 2015

# Sommario

- 1 Correzione esercizi
- 2 Iterazione indeterminata
  - Sintassi ed Esempi
- 3 Esercizi
- 4 Esercizi per casa

## Esercizi a casa

- Scrivere una funzione che presi due numeri come parametri della funzione mi stampi il MCD tra i due numeri. Usare ciclo for.
- Scrivere una funzione che preso un numero come parametro restituisca True se è primo, e False altrimenti. Usare ciclo for.
- Scrivere una funzione che preso un numero come parametro mi stampi tutti i numeri primi che lo dividano. Usare ciclo for.
- Scrivere una funzione che prese in input due stringhe le stampi ordinate in ordine alfabetico.



# Soluzioni

```
import math
def MDC(a,b):
    if a<b:
        t=a
    else:
        t=b
    for i in range((t+1),0,-1):
        if (a%i==0) and (b%i==0):
            return i
def primo(a):
    for i in range(2,int(math.sqrt(a))+1):
        if (a%i==0):
            return False
    return True
```



# Soluzioni

```
def es3(a):  
    d=()  
    for i in range(1,a+1):  
        if (a%i==0) and primo(i):  
            d=d+(i,)  
    return d  
def ordine(a,b):  
    if type(a)==str and type(b)==str:  
        if (a>b):  
            print(b,a)  
        else:  
            print(a,b)  
    else:  
        print("non hai inserito delle stringhe")
```



# Outline

- 1 Correzione esercizi
- 2 Iterazione indeterminata**
  - Sintassi ed Esempi
- 3 Esercizi
- 4 Esercizi per casa

# Iterazione Sintassi

---

```
while <condizione>:  
    <istruzioni_while>
```

---

- **while**: indica l'operatore di iterazione indeterminata.
- **condizione**: è l'espressione booleana che viene controllata la prima volta che si incontra l'istruzione **while** e ogni volta che si conclude la sequenza di **<istruzioni\_while>**
- **istruzioni\_while**: sono l'insieme di istruzioni che vengono eseguite se e soltanto se risulta **True** la condizione

Attenzione:

- Se si usa l'iterazione indeterminata bisogna verificare che l'espressione presente nella condizione venga modificata all'interno del blocco **<istruzioni\_while>**



# Esempio 1

```
def menu():
    print('Seleziona 1 per calcolare le soluzioni della disequazione')
    print('Seleziona 2 per calcolare l''equazione di secondo grado')
    print('Seleziona 3 per uscire')
    x= int(input('Digita la tua scelta'))
    while 1<=x<=3:
        elif x==1:
            a,b,c=diseq_sec_grad_input()
            print(diseq_sec_grado(a,b,c))
            x= int(input('Digita la tua scelta'))
        elif x==2:
            a,b,c=equazione_sec_grad_input()
            print(equazione_sec_grad(a,b,c))
            x= int(input('Digita la tua scelta'))
        elif x==3:
            break
```





# Break

- `break`: si trova annidato in un ciclo (for while). Interrompe l'esecuzione del ciclo



## Esempio 2

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

- Cosa computa questa funzione?



# Esercizio 1

- Scrivi un programma equivalente a quello presentato nell'Esempio 2 utilizzando il ciclo for

# Soluzione esercizio 1

```
def funfor(s):  
    r=''  
    for i in range(len(s)):  
        r=r+s[len(s)-1-i]  
    return r
```



## Esercizio 2

- Scrivi un programma equivalente a quello che segue utilizzando il ciclo while.

---

```
def divisori(x):  
    div=()  
    for i in range(1,x+1):  
        if x%i==0:  
            div=div+(i,)  
    return div
```

---



# Soluzione Divisori

## while

```
def divisori(x):  
    i=1  
    div=()  
    while i<=x :  
        if x%i==0:  
            div=div+(i,)  
        i+=1  
    return div
```

- i viene inizializzata a 1 (primo divisore di ciascun numero)
- si controlla che i sia minore o uguale a x, nel caso sia verificata la condizione
- si verifica la condizione di divisore e in caso affermativo si aggiunge il divisore alla variabile div
- si incrementa i



# Esercizi

Scrivere e commentare un programma con un menu di selezione dove:

- se si seleziona 1: la funzione prende in input una tupla di numeri e restituisca la media dei valori contenuti nella tupla
- se si seleziona 2: la funzione prende in input due tuple e restituisca la differenza delle due tuple  $(1,3,6) - (1,2,2) = (0,1,4)$  solo se le tuple hanno lunghezza uguale
- se si seleziona 3: la funzione prende in input 3 punti rappresentati da tre tuple  $(x_1, y_1), (x_2, y_2), (x_3, y_3)$  e restituisce True se questi punti giacciono sulla stessa retta False altrimenti
- se si seleziona 4 e soltanto 4: il programma esce



# Esercizio

- 1 Si scriva la funzione iterativa che preso come argomento una sequenza restituisca la sequenza dove tutti gli elementi adiacenti uguali siano stati ridotti a un solo elemento. Esempio:  
`rimdup((1,2,3,3,3,2,4,4,3)) → (1,2,3,2,4,3)`
- 2 Scrivere una funzione che presi come parametri due tuple rappresentanti i valori di due vettori restituisca la somma tra i due vettori se è possibile eseguire tale operazione.
- 3 Scrivere una funzione che presi come parametri due tuple rappresentanti i valori di due vettori restituisca il prodotto scalare tra i due vettori se è possibile eseguire tale operazione.





# Soluzione 1

---

```
def rimdup(s):  
    if len(s) <= 1:  
        return s  
    else:  
        o = ()  
        conf = s[0]  
        o = o + (conf,)  
        for i in range(1, len(s)):  
            if conf != s[i]:  
                conf = s[i]  
                o = o + (conf,)   
        return o
```



# 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 una 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 una 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

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



# Cosa abbiamo fatto?

- 1 Correzione esercizi
- 2 Iterazione indeterminata
  - Sintassi ed Esempi
- 3 Esercizi
- 4 Esercizi per casa