

Programmazione

Funzioni

Dario Pescini¹

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi

`dario.pescini@unimib.it`

Utilità Funzioni: esempio

Costruire un programma che calcoli il coefficiente binomiale

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Utilità Funzioni: esempio

Costruire un programma che calcoli il coefficiente binomiale

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Esempio

Prima costruisco una funzione che mi calcoli il valore di $n!$:

$$n! = n (n - 1) (n - 2) \dots 2 1$$

Esercizio - fattoriale

```
n = input("inserire il valore di n ")

i = n
fattoriale = 1
while i > 1:
    fattoriale = fattoriale * i
    i = i - 1

print "\n %s! = " % n, fattoriale
```

Esercizio - binomiale

```
def fattoriale(numero):  
    i = numero  
    prodotto = 1  
    while i > 1:  
        prodotto = prodotto * i  
        i = i - 1  
    return prodotto  
  
print "inserire i valori n e k"  
n = input("n ")  
k = input("k ")  
  
binomiale = fattoriale(n) / (fattoriale(k) * fattoriale(n-k))  
  
print "il coefficiente binomiale è ", binomiale
```

Esercizio - binomiale

```
import math as m

print "inserire i valori n e k"
n = input("n ")
k = input("k ")

binomiale = m.factorial(n) / (m.factorial(k) *
    m.factorial(n-k))

print "il coefficiente binomiale è ", binomiale
```

Esercizio - binomiale

```
import math as m

print "inserire i valori n e k"
n = input("n ")
k = input("k ")

binomiale = m.factorial(n) / (m.factorial(k) *
    m.factorial(n-k))

print "il coefficiente binomiale è ", binomiale
```

Modulo math

Esercizio

Scrivere una funzione che conti tutte le occorrenze di una lettera in una stringa inserita dall'utente, sapendo che è possibile aggiungere un elemento al dizionario tramite l'istruzione:

```
dizionario[chiave] = valore
```

Esercizio

Scrivere una funzione che conti tutte le occorrenze di una lettera in una stringa inserita dall'utente.

```
parola = raw_input('inserire una parola ')
# con la sintassi sotto la parola va racchiusa tra apici
# parola = input('inserire una parola racchiusa tra apici')

print parola

lettere = {}
i = 0
while i < len(parola):
    if parola[i] in lettere:
        lettere[parola[i]] += 1
    else:
        lettere[parola[i]] = 1
    i += 1

for lettera in lettere:
    print lettera, lettere[lettera]
```

Esercizio

Scrivere un programma che individui tutte le occorrenze di una lettera all'interno di una stringa entrambe inserite dall'utente

Esercizio

```
def trova_lettera(word, substring):  
    indice = word.find(substring)  
    return indice  
  
parola = raw_input('inserire una parola ')  
lettera = raw_input('inserire la lettera da cercare ')  
  
posizione = 0  
indici = []  
while posizione < len(parola):  
    nuovo_indice = trova_lettera(parola[posizione:], lettera)  
    if nuovo_indice != -1:  
        indici += [nuovo_indice + posizione]  
        posizione += nuovo_indice + len(lettera)  
    else:  
        posizione = len(parola)  
  
if indici != []:  
    for indice in indici:  
        print indice  
else:  
    print "lettera non trovata"
```