

## esercizi-esame

June 4, 2023

Dato un array di valori, trovare il valore massimo e il valore minimo.

```
[ ]: l = [7, 10, 25, 2, 3, 28, 100, 78, 90]

# trovo il minimo

min = l[0] # inizializzo un minimo - va bene qualunque elemento della lista

for i in l:
    if i < min:
        min = i

print(f"min = {min}")

# l'idea è quella di trovare in n passi (il for scorre una volta sola la lista,
# di n elementi) il minimo,
# inizializzandone uno, confrontando ogni elemento della lista col minimo,
# inizializzato e, se risulta
# inferiore (if), farlo diventare il nuovo minimo fino a che non c'è più un
# numero più piccolo del minimo trovato

# trovo il max

max = l[0]

for i in l:
    if i > max:
        max = i

print(f"max = {max}")

# lo stesso discorso vale per il massimo
```

```
min = 2
max = 100
```

Dato un array di valori, riordinarlo.

```
[ ]: l = [7, 10, 25, 2, 3, 28, 100, 78, 90]

# due metodi:
# 1) bubble sort - è di complessità  $n^2$ , perché fa  $n$  volte  $n$  passi;
# la complessità, legata all'efficienza, è il numero di passi necessario a
    ↳ compiere l'algoritmo

for i in range(len(l)): # i serve solo a ripetere lo scorrimento n volte
    for j in range(len(l)-1): # j e i scorrono la posizione, non l'elemento
        if l[j] > l[j + 1]:
            l[j], l[j + 1] = l[j + 1], l[j] # doppio assegnamento che scambia i
            ↳ valori

print(l)
```

[2, 3, 7, 10, 25, 28, 78, 90, 100]

Scrivere un programma con cui eseguire il metodo di bisezione.

```
[ ]: def f(x):
    y = x**3 + 1
    return y

a = -5
b = 5

while (b - a) > 0.1:
    if f(b)*f(a) < 0:
        c = (b - a)/2 + a
    else:
        print("error")
        break
    if f(c)*f(a) < 0:
        b = c
    else:
        a = c

print(a, b)
```

-1.015625 -0.9375

Scrivere un programma per eseguire un fattoriale.

```
[ ]: def factorial(n):
    if n == 1:
        r = 1
    else:
        r = n * factorial(n-1)
    return r
```

```
print(factorial(3))
```

6

Scrivere un programma per trovare i punti che finiscono dentro una circonferenza.

```
[ ]: import numpy as np

punti = [] # lista ( in cui andrò a mettere le tuple (a, b) )

n = 10

for i in range(n):
    a = np.random.rand()
    b = np.random.rand()
    if np.sqrt(a**2 + b**2) < 1:
        punti.append((a, b))

print(punti)
print(len(punti))
```

```
[(0.03309072947813596, 0.27166689923176424), (0.14819467635236894,
0.0177557337396973), (0.34595081550157036, 0.6821114826583398),
(0.5419186768866701, 0.6193348147329644), (0.19971069907886008,
0.4320696289841933), (0.3254423561021298, 0.9295023291516458),
(0.24753835889479148, 0.7447249279409349), (0.4564127906900891,
0.6513354254979113)]
```

8

Calcolare la media e la deviazione standard della media.

```
[ ]: # media

x = [1, 2, 3]
m = 0

for i in range(len(x)):
    m += x[i]
m /= len(x)

s=0

for i in range(len(x)):
    s += (x[i] - m)**2

s = np.sqrt(s / (len(x) * (len(x) - 1 )))
```

```
print(m)
print(s)
```

2.0

0.5773502691896257

Scrivere una somma geometrica.

```
[ ]: N = 100
      x = 2
      somma = 0

      for k in range(N+1):
          somma += x**k

      print(somma)
```

2535301200456458802993406410751