

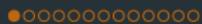
# Programmazione in Python

## While - Esercizi

Dario Pescini

Università degli Studi di Milano-Bicocca

Dipartimento di Statistica e Metodi Quantitativi



# Outline

## 1 Strutture di controllo

# variabili di accumulo: media

Spesso si utilizzano i cicli per “accumulare” quantità variabili in una variabile detta d'accumulo o accumulatore.

```
numIterazioni = 5
media = 0
contatore = 0
while contatore < numIterazioni:
    valore = float(input('inserisci un numero '))
    media += valore
    contatore += 1

media /= numIterazioni
print 'il valor medio è ', media
```

## Esercizio - varianza

Scrivere un algoritmo per calcolare media e varianza di 10 valori inseriti da tastiera dall'utente.

## Esercizio - varianza

Scrivere un algoritmo per calcolare media e varianza di 10 valori inseriti da tastiera dall'utente.

- $\mu = \frac{\sum_{i=1}^N x_i}{N}$
- $\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$

## Esercizio - varianza

Scrivere un algoritmo per calcolare media e varianza di 10 valori inseriti da tastiera dall'utente.

- $\mu = \frac{\sum_{i=1}^N x_i}{N}$
- $\sigma^2 = \frac{\sum_{i=1}^N (x_i - \mu)^2}{N}$
- $\sigma^2 = \frac{\sum_{i=1}^N x_i^2 - \frac{(\sum_{i=1}^N x_i)^2}{N}}{N - 1}$

# Esercizio

```
N = 10

somma = 0.0
somma_quadrati = 0.0

contatore = 0
while contatore < N:
    numero = input('inserisci un numero ')
    print "iterazione ", contatore
    somma += float(numero)
    somma_quadrati += float(numero)**2
    contatore += 1

media = somma / N
varianza = (somma_quadrati - (somma**2 / N)) / (N - 1.0)

print "media = ", media
print "varianza = ", varianza
```

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

Come posso verificarlo?

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

Come posso verificarlo?

- un numero  $n$  è divisibile per un divisore  $d$  se  $n \bmod d = 0$  o equivalentemente  $resto = 0$

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

Come posso verificarlo?

- un numero  $n$  è divisibile per un divisore  $d$  se  $n \bmod d = 0$  o equivalentemente resto  $= 0$
- ha come unici divisori 1 e se stesso o equivalentemente l'insieme dei divisori  $\mathcal{D} \subseteq \mathbb{N}$  deve essere  $\mathcal{D} = \{1, n\}$

## Esercizio

Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

Come posso verificarlo?

- un numero  $n$  è divisibile per un divisore  $d$  se  $n \bmod d = 0$  o equivalentemente resto  $= 0$
- ha come unici divisori 1 e se stesso o equivalentemente l'insieme dei divisori  $\mathcal{D} \subseteq \mathbb{N}$  deve essere  $\mathcal{D} = \{1, n\}$
- posso generare tutti i numeri  $d \in \mathbb{N} \mid d \leq n$  e verificare per ciascuno se  $d$  è divisore

## Esercizio

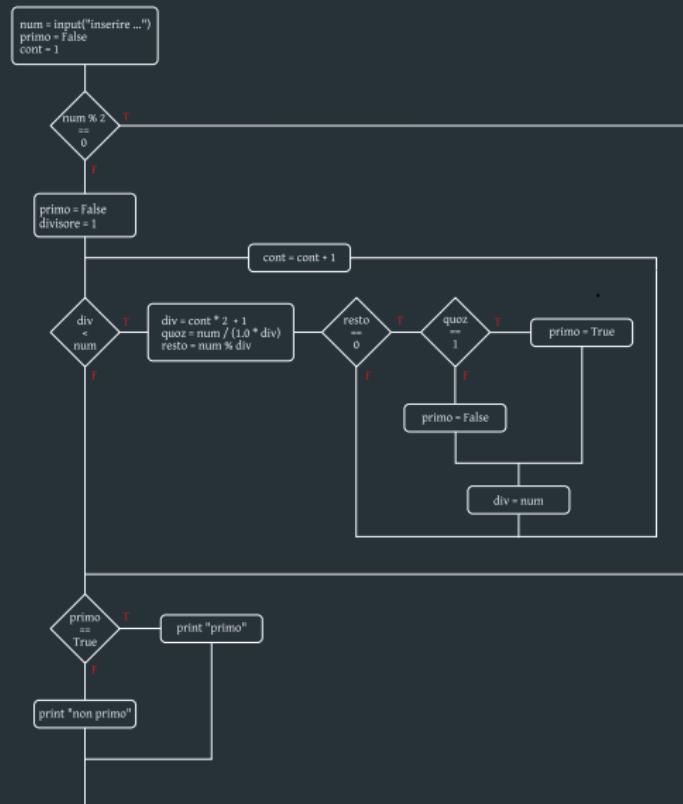
Si costruisca un programma che ricevuto un numero intero riconosca se è primo.

Un numero naturale, maggiore di 1, si dice primo se è divisibile solo per se stesso e per l'unità.

Come posso verificarlo?

- un numero  $n$  è divisibile per un divisore  $d$  se  $n \bmod d = 0$  o equivalentemente resto  $= 0$
- ha come unici divisori 1 e se stesso o equivalentemente l'insieme dei divisori  $\mathcal{D} \subseteq \mathbb{N}$  deve essere  $\mathcal{D} = \{1, n\}$
- posso generare tutti i numeri  $d \in \mathbb{N} \mid d \leq n$  e verificare per ciascuno se  $d$  è divisore
- al primo divisore diverso da  $n$  che incontro,  $n$  non è primo.  
Devo verificare che  $\frac{n}{d} = 1$

# Esercizio

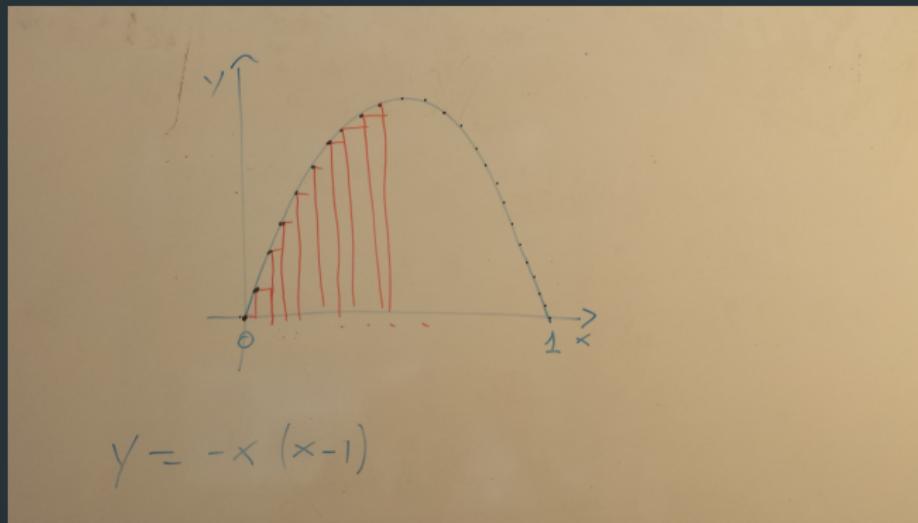


# Esercizio

```
numero = input("inserire un numero intero positivo: ")
primo = False
contatore = 1

if numero%2 == 0:
    primo = False
    divisore = numero
    print numero, "e' un numero pari"
else:
    divisore = 1.0
    while divisore < numero:
        divisore = contatore * 2 + 1
        quoziente = numero / (1.0 * divisore)
        resto = numero % divisore
        print "numero = ",numero,"contatore = ",contatore
        print "divisore = ",divisore,"quoziente = ",quoziente,"resto = ",resto,"\\n"
        if resto == 0:
            if quoziente == 1.0:
                primo = True
                divisore = numero
            else:
                primo = False
                divisore = numero
        contatore = contatore + 1
if primo:
    print numero, "e' un numero primo"
else:
    print numero, " non e' un numero primo"
```

## Esercizio



Si calcoli l'area sottesa dalla parabola in figura.

## Esercizio: decomporre il problema in problemi più semplici

- L'area sottesa può essere approssimata come la somma delle aree di rettangoli contigui aventi per base l'asse delle ascisse e per altezza il punto di contatto con la parabola
- costruire i rettangoli
- determinare base e altezza di un rettangolo
- calcolare area di un rettangolo
- sommare i contributi

# Esercizio: decomporre il problema in problemi più semplici

- L'area sottesa può essere approssimata come la **somma** delle **arie** di **rettangoli** contigui aventi per base l'asse delle ascisse e per altezza il punto di contatto con la parabola
- costruire i rettangoli
- determinare base e altezza di un rettangolo
- calcolare area di un rettangolo
- sommare i contributi

## Esercizio: decomporre il problema in problemi più semplici

- per costruire la base dei rettangoli si può suddividere l'intervallo  $I = [0, 1]$  in un numero finito,  $M$ , di intervalli contigui  $I_i$ .
- in questo caso ciascun intervallo  $I_i$  avrà lunghezza pari a  $\frac{1}{M}$  (base del rettangolo)
- ciascun rettangolo ( $i$ ) avrà per base il segmento  $[\frac{i}{M}, \frac{i+1}{M}]$
- l'altezza potrebbe essere  $y_i = -x_i(x_i - 1) = -\frac{i}{M}(\frac{i}{M} - 1)$

## Esercizio: soluzione

```
numIntervalli = input('inserire il numero di intervalli in  
[0.0, 1.0] ')  
  
deltaIntervallo = 1.0 / float(numIntervalli)  
print "larghezza intervallo", deltaIntervallo  
  
areaSottesa = 0.0  
i = 0  
while i < numIntervalli:  
    xIntervallo = i * deltaIntervallo  
    yIntervallo = - xIntervallo * (xIntervallo - 1.0)  
    areaSottesa += yIntervallo * deltaIntervallo  
    i += 1  
  
print "l'area sottesa dalla curva vale ", areaSottesa
```