

*Università degli Studi di Trieste*

Corso di ingegneria industriale

**Esercitazioni di Fondamenti di Informatica**

Giacomo Strangolino

mailto/chat: [delleceste@gmail.com](mailto:delleceste@gmail.com)

<http://www.giacomos.it>

# Lezione 3 (4/11/2013)

## Ripasso: operazioni sui dizionari

- Operatore [ ]
- *pop(chiave)*: rimuove la chiave, se presente  
*d = {'a':1, 'b':2}*  
*d.pop('a')*
- *clear()*, *copy()*
- *len(dizionario)* # len è funzione **builtin**!
- *for key in d.keys():* # ciclo su chiavi di un dizionario  
*print(key)*
- *for value in d.values():* # ciclo su valori di un dizionario  
...

# Lezione 3

- Si scriva una funzione che, presi in ingresso due interi, ne calcoli il Massimo Comune Divisore, utilizzando l'algoritmo di Euclide
- **Lemma 1.** Siano  $a, b, c \in \mathbf{Z}$ . Se  $c$  divide  $a$  e  $b$  allora  $c$  divide ogni combinazione lineare di  $a$  e  $b$  a coefficienti in  $\mathbf{Z}$ .
- **Lemma 2.** Siano  $a, b, q, r \in \mathbf{Z}$  tali che  $a = bq + r$ . Allora i divisori comuni ad  $a$  e  $b$  sono tutti e soli i divisori comuni a  $b$  e  $r$ . In particolare, i massimi comuni divisori tra  $a$  e  $b$  sono precisamente i massimi comuni divisori tra  $b$  e  $r$ .
- Effettuo la divisione tra  $a$  e  $b \implies a = bq + r$ ,  $r < b$  supposto  $a \geq b$  (altrimenti li posso scambiare)
- $\text{MCD}(a, b) = \text{MCD}(b, r)$ ,

# Lezione 3

- Scrivere una funzione che, data la matrice completa che rappresenta un sistema lineare, ne trovi l'insieme delle soluzioni con il metodo di Gauss-Jordan.
- Si supponga che la matrice fornita rappresenti già un sistema determinato

# Lezione 3

Un sistema lineare di m equazioni in m incognite

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\ &\vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \cdots + a_{m,n}x_n &= b_m \end{cases}$$

Può essere descritto dalla sua **matrice completa dei coefficienti** del sistema:

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & \cdots & a_{m,n} & b_m \end{pmatrix}$$

Per il nostro esercizio supponiamo che i coefficienti del sistema lineare appartengano a  $\mathbf{R}$   
Se con  $\mathbf{A}$  indico la matrice dei coefficienti, con  $\mathbf{B}$  il vettore colonna dei termini noti e con  $\mathbf{x}$  il vettore colonna delle incognite, allora il sistema si può scrivere come

$$\mathbf{Ax} = \mathbf{B}$$

# Lezione 3

**Per risolvere un sistema lineare si può:**

- scambiare l'ordine di scrittura di due equazioni;
- moltiplicare entrambi i membri di un'equazione per un numero diverso da zero;
- sommare ad ogni membro di un'equazione la stessa quantità a sinistra e a destra.

# Lezione 3

## **Sulla matrice, ciò equivale a:**

- scambiare tra loro 2 righe;
- moltiplicare una riga per uno scalare non nullo;
- sostituire una riga con quella ottenuta sommando ad essa un multiplo di un'altra riga.

## **Obiettivo:**

- Ottenere una matrice  $A$  a diagonale unitaria;



Il vettore  $B$  conterrà allora le soluzioni del sistema

# Lezione 3

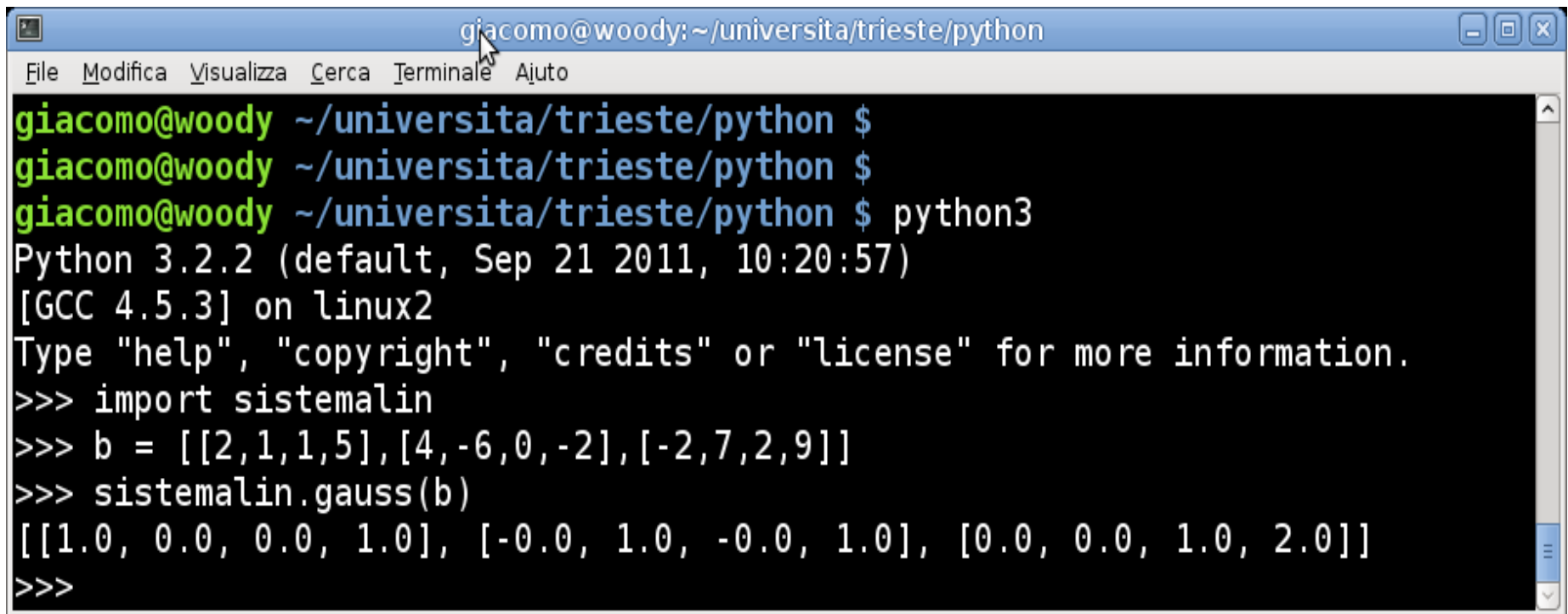
$$\begin{cases} 2x + y + z = 5 \\ 4x - 6y = -2 \\ -2x + 7y + 2z = 9 \end{cases}$$

$$\begin{bmatrix} 2 & 1 & 1 & 5 \\ 4 & -6 & 0 & -2 \\ -2 & 7 & 2 & 9 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 5 \\ 0 & -8 & -2 & -12 \\ 0 & 8 & 3 & 14 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 1 & 1 & 5 \\ 0 & -8 & -2 & -12 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

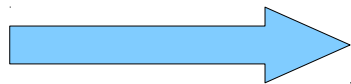
E così via... annullando successivamente anche gli elementi sopra la diagonale e finalmente rendendo la diagonale unitaria



# Lezione 6

A terminal window titled 'giacomo@woody:~/universita/trieste/python' with a menu bar (File, Modifica, Visualizza, Cerca, Terminale, Aiuto). The terminal shows a user running 'python3' which starts Python 3.2.2. The user imports 'sistemalin' and defines a matrix 'b'. Then they call 'sistemalin.gauss(b)' and see the output matrix.

```
giacomo@woody ~/universita/trieste/python $  
giacomo@woody ~/universita/trieste/python $  
giacomo@woody ~/universita/trieste/python $ python3  
Python 3.2.2 (default, Sep 21 2011, 10:20:57)  
[GCC 4.5.3] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> import sistemalin  
>>> b = [[2,1,1,5],[4,-6,0,-2],[-2,7,2,9]]  
>>> sistemalin.gauss(b)  
[[1.0, 0.0, 0.0, 1.0], [-0.0, 1.0, -0.0, 1.0], [0.0, 0.0, 1.0, 2.0]]  
>>>
```



**Soluzioni:  $x = 1.0$   $y = 1.0$   $z = 2.0$**

# Lezione 3

## ***Homework***

- Scrivere una funzione che trasformi un numero intero da base 2 a base 10.

Nota: usare la funzione *builtin* **`ord()`** per ottenere il codice ASCII del carattere passato come argomento.

# Lezione 3

## *Homework*

- Si scriva una funzione che restituisce in un vettore la scomposizione in fattori primi di un numero fornito come parametro
- **NOTA:** ogni numero non può avere più di un divisore grande almeno quanto la sua radice per cui posso cercare un divisore fino alla radice quadrata del numero

# Lezione 3

## *Homework*

- Scrivere una funzione che trasformi un numero intero da base 10 a base 2.
- Scrivere un programma che, dato un numero in base 10, lo rappresenti in complemento a 2 su  $N$  bit (si usi la funzione `base10->base2` scritta sopra per la rappresentazione binaria e l'operatore XOR bit a bit per il complemento a 2).
- Scrivere una funzione che ordini un vettore di interi.

# Lezione 4 (06/11/2013)

- Ricerca sequenziale di un numero in una lista di numeri (le slide delle lezioni 1-2 riportano uno schema a blocchi analogo a quello visto in aula)
- Problema della copia superficiale (copia dei soli riferimenti) e copia profonda (copia degli oggetti). (*deepcopy.py* riporta un esempio per una lista di liste (matrice))
- *deepcopy.py* tratta anche il caso di annidamento di livello maggiore di 2 mediante ricorsione (sarà trattato nelle prossime lezioni)

# Lezione 4 (06/11/2013)

- Gestione degli errori mediante blocchi **try/except**

```
import sys
try:
    f = open('myfile.txt') # apro un file in lettura
    s = f.readline()       # leggo la prima linea del file
    i = int(s.strip())     # cerco di convertirla in un
intero
except IOError:
    print("Errore di lettura dal file 'myfile.txt'")
except ValueError:
    print("Il numero ", s, "non e` un intero!")
except:
    print("Errore sconosciuto")
```

# Lezione 4 (06/11/2013)

- Gestione degli errori: pagina documentazione python di riferimento:

<http://docs.python.org/3/tutorial/errors.html>

# Lezione 4

## Ricerca binaria all'interno di un vettore ordinato

- Scrivere una funzione che, dato in ingresso un vettore *ordinato* e un numero a piacere, dica se il numero è contenuto nel vettore oppure no



# Lezione 4

## Ricerca binaria all'interno di un vettore ordinato

- l'idea è quella di iniziare la ricerca dall'elemento centrale, cioè a metà del vettore (si pensi alla ricerca di una parola in un vocabolario).
- Si **confronta** questo elemento con quello cercato:
- se **corrisponde**, la ricerca termina con successo;
- se è **inferiore**, la ricerca viene ripetuta sugli elementi precedenti (ovvero sulla prima metà del vettore), scartando quelli successivi;
- se invece è **superiore**, la ricerca viene ripetuta sugli elementi successivi (ovvero sulla seconda metà del vettore), scartando quelli precedenti.
- Quando tutti gli elementi sono stati scartati, la ricerca termina indicando che il valore non è stato trovato.

# Lezione 4

## ***Homework***

- Scrivere una funzione atta a calcolare le radici di un'equazione di secondo grado
- Scrivere una funzione che, data una stringa in ingresso, conti il numero di linee, parole, caratteri letti.
- Scrivere una funzione *checkdate(g, m, a)* che controlla la validità della data passata come parametro.
- Scrivere una funzione che inverte gli elementi di una lista di interi. Si usi la funzione python *reverse()* su una lista come esempio e test della funzione scritta.
- Scrivere una funzione per riempire una matrice NxN a spirale, con valori crescenti da 0 a  $N^2 - 1$  iniziando dall'angolo superiore sinistro. La funzione riceve come parametro N.

# Lezione 5

## *Homework*

Esempio matrice a spirale 5x5

0	1	2	3	4
15	16	17	18	5
14	23	24	19	6
13	22	21	20	7
12	11	10	9	8

