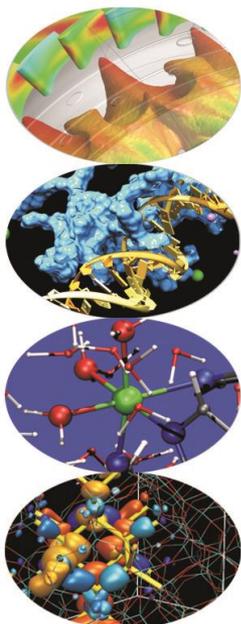
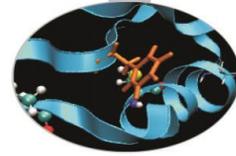


# Esercitazione I/O



# Esercizi I/O



## Esercizio 1

Dato un generico file di dati input.dat: aprire il file in modalità di sola lettura e scriverne a video il contenuto. Contare le occorrenze del carattere \$.

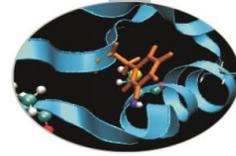
## Esercizio 2

Dato un generico file di dati output.dat: aprire il file in modalità di sola scrittura e passargli i dati da tastiera, finchè non viene digitato il carattere \$.

## Esercizio 3

Dato un file di testo in formato xyzdata.dat

```
100
B -9.71868624e-04 -3.06242577e-03 7.12290507e-03
Si 2.70788718e+00 2.71177311e+00 1.97612206e-03
Si 2.71216726e+00 4.25709686e-03 2.70758408e+00
Si 4.78106353e-03 2.70830095e+00 2.71046762e+00
....
2.72549970e+01 0.00000000e+00 0.00000000e+00
0.00000000e+00 2.72549970e+01 0.00000000e+00
0.00000000e+00 0.00000000e+00 2.72549970e+01
```



# Esercizi I/O

La prima riga contiene il numero di atomi (int) N

La seconda riga è vuota

Le righe seguenti contengono il simbolo atomico e le tre coordinate x,y,z

Le ultime tre righe contengono il tensore (tre elementi per riga)

- Richiedere da standard input il nome del file di input (cout/cin)
- Definire una struct Atom costituita da una stringa e da tre double per le coordinate
- Creare un array di Atom di dimensione N e una array di array per salvare il tensore tensor
- Leggere il file con l'operatore << e immagazzinare in Atom e in tensor i valori letti
- Stampare a video gli atomi letti usando la funzione cout e setw per formattare l'output
- Stampare a video il tensore usando la funzione cout e formattando l'output con right, fixed, setprecision