

# Informatica-Laboratorio

## Settimana 5

Dario Tamascelli

October 28, 2016

### Esercizio 1

Implementare una funzione `float randUnif(float a, float b)` che generi numeri “a caso” (uniformemente distribuiti) nell’intervallo  $[a, b]$ .

### Esercizio 2

Estenderemo qui l’Esercizio 1 svolto la scorsa settimana. Simuleremo quindi un moto rettilineo uniforme in cui, però, le misure di posizione siano affette da errore. La il tempo  $t_{max}$  e il passo  $dt$  di simulazione del moto, inoltre, potranno essere scelti a runtime. Scrivere quindi un programma che:

1. Chieda all’utente la posizione iniziale  $x(t = 0) = x_0$  e la velocità del punto materiale.
2. Chiedi all’utente il tempo finale `tMax` e il “passo di integrazione”  $dt$ , determini il numero di passi `nsteps` di simulazione. Attenzione a non perdere pezzi in giro: al massimo fate un passo di simulazione in più.
3. Allochi un vettore di `float` di dimensione `nsteps`.
4. Determini la posizione registrata al passo  $i$ -esimo della simulazione come:

$$x(t_i) = x_0 + v \cdot i * dt + \varepsilon_i$$

dove  $\varepsilon_i$  è un numero estratto a caso da una distribuzione uniforme in  $[-0.3, 0.3]$ .

5. Calcoli la media e la deviazione standard del campione delle velocità  $v_i$  definite come:

$$v_i = v(t_i) = \frac{x(t_{i+1}) - x(t_i)}{dt}.$$

6. Controlli se la media calcolata al punto precedente coincide con la velocità media:

$$v_m = \frac{x(t_{nsteps}) - x(t_0)}{dt \cdot nsteps}.$$

### Esercizio 3

Scrivere un programma che, caricato un vettore di `float` di dimensione arbitraria (non esagerate che la memoria costa!) e caricato come preferite, ne ordini le componenti in ordine non decrescente (per intenderci  $v_i \leq v_{i+1}$ ). A tal fine potrebbe essere utile usare le funzioni “trovaMin” e “swapArray” che dovrete aver già implementato (o scopiazzato).

IDEA:

- al primo passo, determinare il minimo del vettore e la sua posizione (se ci sono più occorrenze del valore minimo, prendetene una...). Spostare il minimo in testa.
- Al secondo passo iterare quanto fatto al primo passo, ma senza considerare questa volta la prima componente del vettore, che è già a posto, ecc....
- Avrete finito quando il vettore residuo da controllare ha un solo elemento.