

Informatica-Laboratorio

Settimana 4

Dario Tamascelli

October 19, 2016

Esercizio 1

Simuliamo il moto di una punto materiale che si muove di moto rettilineo uniforme a velocità $v = 1.5 \text{ cm/s}$, a partire da una posizione $x_0 = -0.5$, rispetto ad un assegnato sistema di riferimento. Simuleremo il moto per $n = 99$ passi, dove l'incremento temporale ad ogni passo è di $dt = 0.2 \text{ s}$. La posizione al tempo t , $x(t)$ è data dalla legge oraria

$$x(t) = x_0 + v \cdot t.$$

1. Preparare variabili di tipo `x0`, `v`, `dt` di tipo `float` e chiedere all'utente i valori da assegnare a queste.
2. Preparare una variabile `array` di `float` di $n = 100$ elementi.
3. Registrare nella prima posizione del vettore il valore x_0 .
4. Registrare nelle rimanenti posizioni del vettore i valori $x(t_i = dt \cdot i) = x_0 + v(dt \cdot i)$.
5. Stampare a video il vettore.

Determinate quindi l'intervallo di tempo in cui la posizione del punto materiale passa da negativa a positiva.

Esercizio 2

Scrivere un programma che, per fissato N :

1. Prepari le variabili `float min`, `max`, `m` e `varC` e il vettore `float dati[N]`.
2. Chieda all'utente quanti dati vuole inserire, verifichi che i dati siano contenibili nel vettore e proceda al caricamento dei dati.
3. Determini la media e la varianza del campione dei valori inseriti.

4. Una volta controllata la correttezza delle operazioni richieste, dichiarare e definire due funzioni:
`float media(float [],int)`
`float varC(float [],int)`
e usarle dove opportuno nel codice.
5. Determini il valore massimo e il valore minimo e la loro posizione nell'array.
6. Effettuare gli opportuni spostamenti nell'array in modo tale che il primo elemento dell'array sia il valore minimo inserito e l'ultimo elemento dell'array sia il valore massimo inserito. Se l'utente ha inserito un numero di valori, chiamiamolo `dim` strettamente minore di N , l'ultimo elemento *valido* dell'array si trova in posizione `dim-1`.

Nel file `swapArrayFloat.h` e `swapArrayFloat.C` troverete, rispettivamente, la dichiarazione e la definizione della funzione che permette lo scambio di due elementi di un array.

Esercizio 3

Definite una *procedura* `divEucl` che prenda in ingresso due interi `a,b` e un vettore `int res[2]`, e, una volta controllato che il divisore sia $\neq 0$ calcoli il quoziente e il resto della divisione euclidea; il quoziente dovrà essere registrato in `res[0]`, il resto in `res[1]`. Se il divisore inserito fosse $= 0$, la procedura dovrà mettere il valore -1 in `res[1]`. Ovviamente la funzione che chiama `divEucl` potrà controllare la presenza di problemi ispezionando il valore del resto. La soluzione del problema della divisione euclidea per numeri interi relativi la troverete nel file `progDiv.C`.