

**Laboratorio di Calcolo**  
**Laurea Triennale in Fisica**  
Esame del 25.03.2004

Si richiede di scrivere un programma funzionante in ANSI/ISO C che risponda ad uno (solo) dei 3 problemi seguenti. Il programma non dovrà utilizzare librerie specifiche di alcuni sistemi operativi e/o compilatori, come ad esempio la libreria `conio.h`. Il programma può essere consegnato su un floppy, su un CD o su una penna USB, che contenga tutti i files necessari a compilare ed eseguire il programma. I files dovranno essere files ASCII scritti utilizzando un editore (ad esempio emacs) e non un word-processor (ad esempio Word) e non dovranno contenere caratteri speciali come è, ù, etc.

Il programma può essere consegnato anche su supporto cartaceo come listato stampato o in forma scritta a mano, purchè leggibile. In questo caso potrebbe essere richiesto al candidato di scrivere il programma su un computer in sede di esame.

Ciascun problema da diritto ad un voto massimo di 24, 27, e 30, come indicato.

**Es. 1**

La quota  $h$  e velocità  $v$  di un punto materiale di massa  $m$  posto inizialmente fermo ad una quota  $h_0$  e soggetto alla sola forza gravitazionale variano in funzione del tempo come

$$h(t) = h_0 - \frac{1}{2} g t^2$$

$$v(t) = -g t$$

dove  $g$  è l'accelerazione di gravità.

Si chiede di scrivere un programma che scriva i valori dell'energia cinetica  $T(t)$  e potenziale  $U(t)$  del punto materiale in funzione del tempo dall'istante iniziale fino al momento in cui tocca terra. Il programma deve inoltre soddisfare le seguenti richieste:

1. I valori di  $h_0$  ed  $m$  devono essere letti dallo `stdin` (tastiera). Il programma deve verificare la ragionevolezza dei dati immessi.
2. La risoluzione temporale  $dt$  da utilizzare deve essere letta dallo `stdin` (tastiera).
3. L'energia cinetica  $T(t)$  e potenziale  $U(t)$  devono essere valutate utilizzando due funzioni separate, una per l'energia cinetica ed una per l'energia potenziale.
4. I risultati devono essere scritti sullo `stdout` (terminale) con il seguente formato

$$t \quad T \quad U$$

ossia ogni riga dell'output deve contenere i tre valori: tempo, energia cinetica ed energia potenziale. Facoltativamente si può utilizzare un file al posto dello `stdout`.

Si ricorda che

$$T(t) = \frac{1}{2} m v(t)^2$$
$$U(t) = m g h(t)$$

[Voto massimo 24]

**Es. 2**

Una successione di  $N > 3$  numeri interi è detta palindroma se la sequenza dei numeri è la stessa sia scorrendo la successione dall'inizio verso la fine che dalla fine verso l'inizio. Ad esempio la successione 123321 è palindroma mentre la successione 123322 non lo è.

Si chiede di scrivere un programma che verifichi se una successione data sia palindroma o no. Il programma deve inoltre soddisfare le seguenti richieste:

1. La sequenza deve essere letta dallo **stdin** (tastiera).
2. Si deve usare un array per memorizzare la sequenza.
3. Il controllo deve essere effettuato utilizzando una funzione che ritorna il valore 1 se la sequenza è palindroma ed il valore 0 nel caso non lo sia.
4. Il programma deve scrivere sullo **stdout** (terminale):

```
sequenza palindroma      ⇒  "E' palindroma"
sequenza non palindroma  ⇒  "Non e' palindroma"
```

[Voto massimo 27]

### **Es. 3**

Il gioco del “campo minato” consiste nel determinare il numero medio di passi che si possono effettuare su una scacchiera prima di incontrare una mina.

Si chiede di scrivere un programma che

1. Definita una scacchiera quadrata  $N \times N$  assegni a ciascuna casella una *mina* con probabilità  $p$ .
2. Posizionato un camminatore al centro della scacchiera, che si assume libero da mine, determini il numero di passi effettuati prima di incontrare una mina. Ad ogni passo il camminatore si può muovere con la stessa probabilità in alto/basso/destra/sinistra.
3. Determini il numero medio di passi effettuati come media sui passi effettuati da  $N_c$  camminatori.

Suggerimenti:

- i) Per generare un evento con probabilità  $p$  si può utilizzare un numero  $x$  con distribuzione uniforme tra  $[0, 1)$ . Se  $x$  è minore di  $p$  l'evento accade altrimenti non accade.
- ii) Per generare un numero  $x$  con distribuzione uniforme tra  $[0, 1)$  si può utilizzare l'istruzione

```
x = (double) rand() / (RANDMAX + 1.0)
```

Per cambiare la sequenza di numeri generati utilizzare, una sola volta, la funzione **srand(seed)** dove **seed** è un numero intero. Per utilizzare le funzioni **rand()** e **srand()** si deve includere il file di header **stdlib.h**.

[Voto massimo 30]