

# Laboratorio di Calcolo per Fisici, Prima prova di esame,

AA 2018-19

Lo scopo della prova di esame è scrivere un programma che simuli il gioco del bowling. Per svolgere l'esercitazione avrete 3 ore; sono concessi libri di testo e appunti, ma **l'uso di cellulari, laptop e tablet non è ammesso, pena l'annullamento del compito.** Il programma va scritto e salvato esclusivamente nella *home directory* dell'utente, utilizzando lo user-id `lcexamXX` (password `lcexamXX`) dove `XX` è il numero della vostra postazione su un file di nome `bowling.c`. Per sicurezza inserite nelle prime righe del file due righe di commento contenenti il vostro nome, cognome e numero di matricola.

► **Esercizio:** Nel gioco del *bowling* i giocatori si affrontano, cercando di realizzare il numero maggiore di punti, abbattendo dei birilli, posti alla fine di una pista. Il gioco si svolge in una sequenza di *frame*, in ciascuno dei quali ogni giocatore ha a disposizione due lanci, con i quali tenta di abbattere dieci birilli. Il punteggio ottenuto in ciascun *frame* è dato dalla somma dei birilli abbattuti nei due lanci; nel caso in cui vengano abbattuti tutti i birilli, il giocatore realizza uno *strike* (se tutti i birilli vengono abbattuti al primo lancio) o uno *spare* (se i birilli vengono abbattuti in due lanci), che danno diritto a dei punti aggiuntivi. Alla fine del numero stabilito di *frame* viene attribuito il punteggio totale, dato dalla somma dei punti di tutti i *frame*.

Si scriva un programma che simuli una partita di bowling di massimo 10 000 *frame* con le seguenti condizioni:

1. I birilli sono rappresentati da un *array* intero di dimensione 10, chiamato `birilli`; un birillo in piedi è rappresentato da un uno (1), un birillo abbattuto da uno zero (0); un secondo *array* di nome `prob`, di dimensione e tipo opportuni, contiene la probabilità di ciascun birillo di essere abbattuto (*vedi sotto*).
2. All'inizio della partita, il programma chiede al giocatore di inserire il numero di frame  $n_f$  di cui è composta, e verifica che si tratti di un numero ammesso.
3. A questo punto, le seguenti operazioni vengono ripetute per ogni frame:
  - Tutti gli elementi dell'array `birilli` sono inizializzati a 1 (cioè tutti i birilli sono in piedi).
  - Una funzione `lancio` simula il lancio della palla da bowling, restituendo il numero di birilli abbattuti. Ciascun birillo è abbattuto con probabilità  $P_a(i)$ , dove  $P_a(i)$  è il valore contenuto nella componente corrispondente dell'array `prob`.
  - Al primo lancio di ciascun *frame*,  $P_a(i)$  è pari al 75% per tutti i birilli; al secondo lancio, la probabilità rimane pari al 75% per i birilli i cui primi vicini (*p.v.*) sono in piedi, e scende al 50% altrimenti. Si definiscono primi vicini di un birillo  $i$  i birilli con indice  $(i - 1)$  e  $(i + 1)$ ; per semplicità, si immagini che la fila di birilli sia chiusa su se stessa (condizioni periodiche al contorno), cioè che i birilli 1 e 10 siano primi vicini tra loro. L'aggiornamento delle probabilità  $P_a(i)$  è svolto da una funzione `aggiorna`.
  - Alla fine di ciascun lancio il programma stampa la situazione dei birilli, utilizzando il seguente formato (il numero N è pari al numero di birilli abbattuti):

| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |    N = 4

- Il punteggio di ciascun lancio è salvato in un array `points`, di dimensione opportuna.
4. **Alla fine della partita**, una funzione `punteggio` legge l'array `points` e calcola il punteggio finale come segue: Per ciascun *frame*, il punteggio è pari al numero di birilli abbattuti. Per uno *strike* si assegnano 10 punti aggiuntivi, per uno *spare* 5 punti aggiuntivi.
  5. Si stampa un riepilogo finale della partita, con il seguente formato:

```
|    10 |    0 | STRIKE!  
|     9 |    1 | SPARE!  
|     9 |    0 |     9  
|     8 |    1 |     9
```

6. Si stampa, inoltre, il numero totale di *strike* totalizzati nel corso della partita.