

Laboratorio di Calcolo per Fisici, Prova di esame,

AA 2018-19

Lo scopo della prova di esame è scrivere un programma che simuli il comportamento di un semplice automa cellulare.

Per svolgere la prova 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 **studente** (password **informatica**) su un file di nome **cell.c**. Per sicurezza inserite nelle prime righe del file due righe di commento contenenti il vostro nome, cognome e numero di matricola.

► **Esercizio:** Gli *automi cellulari* sono modelli matematici discreti studiati in diverse branche della matematica, fisica, informatica e biologia, perché riproducono in maniera schematica il comportamento di diversi sistemi complessi. L'esempio più famoso è probabilmente il *gioco della vita*, introdotto da John Conway negli anni '70.

Un automa cellulare consiste di una griglia regolare di *celle*, ciascuna delle quali può trovarsi in un numero finito di stati discreti, e di una serie di *regole*, che determinano l'evoluzione temporale dello stato di una cella sulla base dello stato delle celle circostanti. In questo esercizio simuleremo il comportamento di un automa cellulare unidimensionale elementare, costituito da una fila di celle, ciascuna dei quali può assumere due valori. Questo significa che a ogni passo t^1 lo stato del sistema è descritto da un vettore del tipo:

| 1 | 0 | 0 | 1 | 1 | 1 | ...

Il comportamento di ciascuna cella al passo $t + 1$ è determinato dalla configurazione delle due celle ad essa adiacenti (a sinistra, e destra, rispettivamente), secondo la seguente *regola*: il valore contenuto nella cella i -sima al passo t rimane immutata al passo $t + 1$ se la cella che precede e quella che segue sono diverse, altrimenti cambia (da 0 a 1 o viceversa). Per semplicità in questo esercizio assumeremo che le celle all'estremo destro e sinistro della sequenza rimangano invariate.

Questo significa, per esempio, che la configurazione specificata sopra si evolverebbe in:

| 1 | 1 | 1 | 0 | 1 | 1 | ...

Si scriva un programma che simuli l'evoluzione temporale dell'automa cellulare, rispettando le seguenti condizioni:

1. Il programma chiede al giocatore di inserire il numero di passi per l'evoluzione temporale dell'automa cellulare (n_t); il valore immesso deve essere compreso tra 1 e 100. Inoltre, chiede al giocatore di inserire un numero intero compreso tra 0 e 30 che rappresenta la probabilità p (in percentuale), con cui vengono inizializzati gli elementi dell'array iniziale (vedi sotto). Se i valori inseriti non sono corretti, il programma reitera la richiesta.
2. lo stato del sistema al passo t è rappresentato da un *array* intero di 50 elementi, chiamato **cell**. **All'inizio della simulazione**, l'array **cell** viene inizializzato in maniera casuale da una funzione **genera**, che assegna a ciascun elemento dell'array il valore 1 con probabilità p , e 0 altrimenti.

¹sebbene rappresenti l'evoluzione temporale, t è l'etichetta del passo ed è dunque adimensionale

3. a questo punto, le seguenti operazioni vengono ripetute **per ogni passo t** :

- Lo stato del sistema si evolve secondo le regola sopra elencate. L'evoluzione è gestita da una funzione **aggiorna**, di tipo e argomenti opportuni. Essa dovr prendere in input l'array **cell**, farne una copia e modificare l'originale in base alla regola di evoluzione.
- Il programma, tramite una funzione **conta**, conta le celle vive (cio con elemento dell'array pari a 1) e stampa sullo schermo un messaggio del tipo:

```
t = 97   celle vive=20
```

- il programma stampa sullo schermo il vettore **cell** aggiornato, con il formato:

```
|1|0|1|0|1|1|1|0|1|0|1|  ...
```

4. **alla fine della simulazione**, il programma stampa un riepilogo, con le seguenti informazioni:

- Numero di passi effettuati.
- Configurazione del vettore **cell** di partenza (che deve dunque essere copiato all'inizio) e finale, con il formato specificato al punto 4.
- Numero di elementi non nulli del vettore di partenza e di quello di arrivo.