

Fondamenti di Informatica 1 – 13-12-08 – Sterbini

NOTA: per ogni esercizio si scriva (oltre all'implementazione) un frammento del codice del main che mostra 1) la definizione delle variabili passate alla funzione e 2) come la funzione viene chiamata.

Esercizio 1

Si scriva la funzione **moda_e_mediana** che:

- riceve come argomenti un vettore di interi V, la sua dimensione N e tutti gli altri argomenti che ritenete necessari.
- calcola il valore della moda del vettore (l'elemento più frequente)
- calcola il valore della mediana del vettore (l'elemento che è maggiore di meno di N/2 elementi e minore di meno di N/2 elementi)
- torna i due valori calcolati in due variabili passate alla funzione per riferimento

Esempio:

Se il vettore V contiene: 1 3 2 6 4 8 9 12 -7 2 5 8 2

La funzione deve tornare i valori: 2 (moda, che appare 3 volte) e 4 (mediana, visto che 4 è maggiore dei 6 elementi -7 1 2 2 2 3 e minore dei 6 elementi 5 6 8 8 9 12)

Esercizio 2

Si supponga di voler implementare il gioco LIFE, in cui, ad ogni passo di tempo, una matrice bidimensionale M di interi di dimensioni RxC si evolve come segue:

- ogni cella può contenere il valore 0 (morta) o 1 (viva)
- la prima colonna della matrice è adiacente all'ultima, e la prima riga è adiacente all'ultima riga (si comporta come la superficie di una ciambella oppure come una mappa di Karnaugh)
- i “vicini” di una cella sono le 8 celle che la circondano più la cella stessa
- una cella è (o diventa) viva se il numero di vicini vivi è 2 oppure 3, altrimenti è (o diventa) morta

Si implementi la funzione **life_step** che:

- riceve come argomenti la matrice M, le due dimensioni R ed C e tutti gli altri argomenti che ritenete necessari
- aggiorna tutte le celle come descritto

NOTA: per evitare errori, l'aggiornamento della matrice non può essere fatto contemporaneamente al calcolo ma si ha bisogno di una matrice uguale di appoggio.

Esempio: se la matrice è

```
0 0 1 0 0 0
0 1 0 1 0 0
0 0 0 0 0 0
0 0 1 0 0 0
```

Dopo l'esecuzione della funzione la matrice sarà

```
0 1 0 1 0 0
0 1 1 1 0 0
0 1 1 1 0 0
0 1 1 1 0 0
```

Esercizio 3

Si supponga di dover confrontare due stringhe (come la funzione strcmp della libreria string.h) per determinare quale delle due viene prima dell'altra nell'ordine lessicografico (quello del dizionario).

Si scriva la funzione **confronta** che:

- riceve come argomenti le due stringhe, di dimensioni sconosciute, terminate da '\0' e tutti gli altri argomenti che ritenete necessari
- torna in due variabili passate alla funzione per riferimento:
 - +1, 0 o -1 per indicare quale delle due stringhe è minore/uguale/maggiore
 - in quale posizione è stata trovato il primo carattere diverso (oppure -1 se sono uguali)
- date della funzione una implementazione **ITERATIVA**

Esempio:

Se le due stringhe sono: “pippo” e “pluto” la funzione deve tornare -1 (“pippo” < “pluto”) e 1 (la posizione del primo carattere diverso)

Esercizio 4

Si implementi la funzione dell'esercizio 3 in modo **RICORSIVO**