

Laboratorio Informatica A - Incontro 4

Problema 1

Scrivere una funzione *f* (e *main* per chiamarla) che riceve due array di 5 interi *A* e *B* e inserisce in ogni casella di *B* la somma degli elementi di *A* dall'inizio alla posizione corrispondente a quella in cui si sta inserendo.

Esempio: *A*=[2,3,4,6,8]

B=[2,2+3,2+3+4,2+3+4+6,2+3+4+6+8] → *B*=[2,5,9,15,23]

Problema 2

Scrivere una funzione che riceve una matrice *M* di interi e un intero *K* e restituisce 1 se la matrice ha una riga i cui valori sono tutti minori di *K*. Si usi il file *lab4ges.c* allegato per scrivere e testare la funzione

Problema 3

Si consideri un videogame dal nome “Godzilla Frenzy”. In tale gioco, il mondo è una griglia *M*×*N* e in ogni cella l'utente può costruire delle case o delle strade. Un esempio di mondo è la seguente griglia

X	0	1	2	3	4	5	6	7	8	9
0	C	C	C	C	C	C	C	C	C	C
1	C	C	C	C	C	C	C	S	C	C
2	S	S	S	S	S	S	S	S	C	C
3	C	S	C	C	C	C	C	S	C	C
4	C	C	S	C	C	C	C	S	C	C
5	C	C	S	S	S	C	C	S	S	S
6	S	S	S	S	S	C	C	S	C	C
7	C	C	S	C	C	S	S	S	C	C
8	C	C	S	C	C	C	C	S	C	C
9	C	C	S	C	C	C	C	S	C	C

Mentre l'utente costruisce la sua città, Godzilla può arrivare improvvisamente e distruggere ciò che è stato costruito. Tuttavia, Godzilla si può muovere solo in orizzontale e in verticale e ha una paura tremenda delle strade e non vuole assolutamente attraversarle. Pertanto, Godzilla atterra su una casella scelta a caso e si può muovere solo sulle case, quando trova una strada (o il bordo della griglia) è costretto ad indietreggiare. Implementare una funzione che permette di calcolare, a partire da una casella di partenza (quella in cui atterra Godzilla), su quali caselle il mostro si può muovere. Ad esempio, se consideriamo la griglia precedente e la casella di coordinate (4,4), la funzione dovrà marcare nella griglia le seguenti caselle

X	0	1	2	3	4	5	6	7	8	9
0	C	C	C	C	C	C	C	C	C	C
1	C	C	C	C	C	C	C	S	C	C
2	S	S	S	S	S	S	S	S	C	C
3	C	S	g	g	g	g	g	S	C	C
4	C	C	S	g	g	g	g	S	C	C
5	C	C	S	S	S	g	g	S	S	S
6	S	S	S	S	S	g	g	S	C	C
7	C	C	S	C	C	S	S	S	C	C
8	C	C	S	C	C	C	C	S	C	C
9	C	C	S	C	C	C	C	S	C	C

Problema 4

Costruite un programma per gestire una lista di film. Per immagazzinare i film dovete usare la seguente struttura:

```
typedef struct _movie {
    char title[200];
    char type[200];
    int year;
} movie;
```

Ovviamente, per immagazzinare i film che vengono inseriti dall'utente dovete usare un array. In questo caso, il dato contenuto in ogni casella dell'array non sarà un semplice *char* o un *int*, ma una struttura di tipo *movie*. Nell'array i nuovi film devono essere inseriti in modo ordinato a seconda dell'anno in cui è stato girato il film (prima i film più vecchi poi quelli più nuovi).

movie *film*[100];

int *numFilm*;

numFilm servirà a sapere quante caselle dell'array sono effettivamente usate

Per questo progetto implementate le funzioni che devono essere richiamate opportunamente in un apposito *main* di test:

- *int* *add*(*movie* *f*[],*int* *numFilm*) //chiede i dati per un nuovo film, lo aggiunge nella posizione corretta e restituisce il numero di film aggiornato.
- *void* *print*(*movie* *f*[],*int* *numFilm*) //stampa la lista dei film.
- *movie* *search*(*movie* *f*[],*int* *numFilm*, *char* * *title*) //cerca un film nella lista in base al titolo e restituisce l'elemento dell'array che contiene tale film.
- *int* *remove*(*movie* *f*[],*int* *numFilm*,*char* * *title*) //rimuove un film senza lasciare buchi e restituisce il numero di film aggiornato.