

LABORATORIO INFORMATICA A - Sessione 4

ESERCIZIO 1 (matrice - non ricorsiva)

Scrivere un programma che richieda all'utente di riempire una matrice 3 x 3 di interi

- la stampi
- sostituisca nella matrice ogni numero dispari con il suo cubo
- stampi la matrice così modificata
- infine, calcoli e stampi la media di tutti gli elementi.

ESERCIZIO 2 (somma di array ricorsiva)

Scrivere una funzione ricorsiva che permetta di sommare due array; il primo array va considerato da 0 a n, mentre il secondo array da n a 0.

ESERCIZIO 3 (abaco ricorsivo)

Scrivere un programma che permetta di visualizzare un qualsiasi numero come un abaco:
eg. 1345

```
0
000
0000
00000
```

Il programma continua a chiedere input fino all'inserimento di 0.

La scomposizione in abaco deve essere eseguita mediante funzione RICORSIVA.

ESERCIZIO 4 (serie di Maclaurin ricorsiva)

Si vuole calcolare $\cos(x)$ mediante lo sviluppo in serie di Maclaurin di ordine n.

Data la formula:

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \quad \text{per ogni } x$$

Scrivere tutte le funzioni ricorsive necessarie per il calcolo dello sviluppo in serie.

ESERCIZIO 5 (strutture dati - lista film)

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 lista_film[100];
int numFilm;
...
```

numFilm servirà a sapere quante caselle dell'array sono effettivamente usate. Per questo esercizio 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.

ESERCIZIO 6 (vette in un array ricorsivo)

Scrivere un programma con funzioni ricorsive che calcoli il numero di “vette” (cioè il numero di elementi che sono maggiori di TUTTI i successivi) in un array.

Sono necessarie due funzioni:

1. Funzione ricorsiva che calcola i massimi
2. Funzione ricorsiva che calcola le vette

eg.

6 4 5 3 1 10 1 10 3 3 → 2

6 4 8 6 10 1 7 2 1 4 → 3

ESERCIZIO 7 (determinante ricorsivo)

Scrivere la funzione ricorsiva che, utilizzando il metodo di Laplace, permetta di calcolare il determinante di una matrice.

$$\det M = \sum_{j=1}^n (-1)^{i+j} m_{ij} \det M_{ij}$$