

# Esercitazione 5

## Argomento: Matrici e File

Istruzioni per questa esercitazione

- Scaricare il file [es5\\_mat.tgz](#) da e-learning.
- Decomprimere il file.
- Aggiungere in [es5\\_mat.c](#) l'implementazione delle funzioni proposte in questa esercitazione.
- Modificare il main() in [es5\\_mat\\_test.c](#) per testare le funzioni che avete implementato. Potete usare le funzioni che trovate già implementate nel file es5\_mat.c per allocare, leggere da file e stampare a video matrici. Usate i file di matrici forniti nell'esercitazione precedente ([mat\\_1.txt](#), [mat\\_2.txt](#), [mat\\_3.txt](#), [mat\\_4.txt](#)) come matrici di test.
- Eseguite l'istruzione **make** nel terminale per compilare. Questa istruzione produrrà vari file, fra cui l'eseguibile es5\_mat\_test.
- **./es5\_mat\_test** per eseguire il main

## Esercizio 5.1

Scrivere la funzione C

```
bool Mat_is_symmetric(Mat *m);
```

che data in ingresso una struttura Mat m contenente una matrice, verifichi che m sia simmetrica o meno. Se m e' simmetrica la funzione deve restituire true in uscita, altrimenti deve restituire false. Si ricorda che una matrice e' simmetrica se ogni elemento  $X_{ij}$  e' uguale all'elemento  $X_{ji}$ .

## Esercizio 5.2

Scrivere la funzione C

```
void Mat_normalize_rows(Mat *m);
```

che, data in ingresso una struttura Mat m contenente una matrice, modifichi m in modo da normalizzare le righe. Si ricorda che la normalizzazione di una riga si ottiene dividendo tutti gli elementi della riga per il modulo della riga stessa.

## Esercizio 5.3

Scrivere la funzione C

```
Mat* Mat_clone(Mat *m);
```

che, data in ingresso una struttura Mat m contenente una matrice, allochi e restituisca una copia della matrice m.

## Esercizio 5.4

Scrivere la funzione C

```
Mat* Mat_sum(Mat *m1, Mat *m2);
```

che, date in ingresso due strutture Mat m1 e Mat m2 contenenti due matrici, allochi e restituisca la somma delle suddette matrici. Nel caso non fosse possibile eseguire la somma (per esempio, se le dimensioni delle due matrici di input non sono uguali), la funzione deve stampare a schermo un messaggio di errore e ritornare NULL.

## Esercizio 5.5

Scrivere la funzione C

```
Mat* Mat_product(Mat *m1, Mat *m2);
```

che, date in ingresso due strutture Mat m1 e Mat m2 contenenti due matrici, allochi e restituisca il prodotto delle suddette matrici. Nel caso non fosse possibile eseguire il prodotto (per esempio, se le dimensioni delle due matrici di input non consentono il prodotto), la funzione deve stampare a schermo un messaggio di errore e ritornare NULL.

## Esercizio 5.6

Scrivere la funzione C che, date in ingresso una struttura Mat m1, allochi e restituisca una struttura Mat m2 contenente tutti elementi con valore 0 o 1. Gli elementi di m2 posti a 1 sono quelli che nella matrice m1 hanno valore massimo tra quelli del loro intorno (di 8 caselle per gli elementi interni, meno per gli elementi sul bordo).

Esempio:

Input: m1

[[1	2	3	3]
[2	13	7	8]
[9	20	6	7]]

Output: m2

[[0	0	0	0]
[0	0	0	1]
[0	1	0	0]]