Esercizi di verifica

5 – Approfondimento sul linguaggio C: Allocazione dinamica della memoria

P2_05_02_C

29. [liv.1] A partire dalla matrice A(m×n), del tipo sotto indicato, allocata per righe
    - staticamente,
    - dinamicamente

    visualizzarne gli elementi per colonne:

$$\mathbf{A}_{4\times 6} = \begin{pmatrix} 11 & 12 & 13 & 14 & 15 & 16 \\ 21 & 22 & 23 & 24 & 25 & 26 \\ 31 & 32 & 33 & 34 & 35 & 36 \\ 41 & 42 & 43 & 44 & 45 & 46 \end{pmatrix}$$

    Gli elementi $a_{i,j}$ della matrice sono tali che le unità indicano la colonna e le decine indicano la riga cui l'elemento appartiene.

P2_05_03_C

30. [liv.1] Scrivere una *function C* che restituisca la matrice C *prodotto righe×colonne* [vedi pdf delle dispense] di due matrici rettangolari A e B le cui dimensioni sono stabilite in input (usare per tutte le matrici l'allocazione dinamica e generarle come numeri reali random). C'è qualche preferenza nell'usare `malloc()` o `calloc()` rispettivamente per A, B o C? Verificare se i tempi di esecuzione, per la sola allocazione e totali, sono gli stessi.

31. [liv.3] Ripetere l'esercizio precedente sul *prodotto righe×colonne di matrici*, una prima volta, allocando tutte le matrici in memoria per colonne ed, una seconda volta, per righe. Per ciascun tipo di allocazione in memoria, scrivere due *function C* per il *prodotto righe×colonne*: una che acceda a tutte le matrici per colonne e l'altra per righe. Confrontare i tempi d'esecuzione delle due modalità di accesso alle matrici rispetto alla loro allocazione in memoria, deducendo quindi il tipo di accesso più efficiente rispetto al criterio di memorizzazione.