**Operazioni Matrici**

**Modifiche apportate per la gestione di una matrice come array monodimensionale**

**Per gestire una matrice mediante un array monodimensionale è necessario cambiare il campo relativo alla struttura dati adoperata per rappresentare la matrice.**

**Prima della modifica:**

**struct matrice**

**{**

**float matrice [righe][colonne];**

**int righe;**

**int colonne;**

**int id;**

**}**

**Post modifica:**

**struct matrice**

**{**

**float matrice [righe\*colonne];**

**int righe;**

**int colonne;**

**int id;**

**}**

**Oppure mediante puntatore singolo**

**struct matrice**

**{**

**float \*matrice;**

**int righe;**

**int colonne;**

**int id;**

**}**

**Nell’ultimo caso considerato prima di poter adoperare nel proprio programma tale struttura dati è necessario procedere nell’allocazione dinamica della memoria che provvederà a metter a disposizione un numero di celle di memoria pari al prodotto delle righe e delle colonne della matrice che si intende rappresentare.**

**Pseudocodice dell'allocazione dinamica:**

**funzione AllocazioneMatrice (matrice)**

**{**

**m -> matrice = malloc ( righe\_matrice \* colonne\_matrice \* sizeof( tipo ) );**

**}**

**Nel codice le funzioni che subiranno le modifiche sono le seguenti:**

**- leggiValore( matrice , i, j );**

**- scriviElemento( matrice , i, j, n ).**

**Le modifiche da apportare al codice sono relative all'accesso ad un elemento della matrice.**

**Prima della modifica: m -> mat [riga] [colonna]**

**Post modifica: m -> mat + ( riga \* numero\_colonne\_matrice + colonna )**

**Pseudocodice delle funzioni interessate:**

**float leggiValore( matrice \*m, int i, int j )**

**{**

**return \*(m->mat+(i\*leggiColonne(m)+j));**

**}**

**void scriviElemento( matrice \*m, int i, int j, float n )**

**{**

**\*(m->mat+(i\*leggiColonne(m)+j)) = n;**

**}**

**Infine, dopo le modifiche sopracitate qualsiasi sia la struttura dati che si adopererà per rappresentare la matrice, il codice e le funzionalità resteranno completamente invariate con la sola differenza che si dovrà omettere l'uso della funzione relativa all'allocazione dinamica nel caso in cui la struttura adoperata sia del tipo: float matrice [righe] [colonne] oppure float matrice [righe \* colonne].**