Libreria per la gestione/manipolazione delle matrici

Una matrice dinamica è caratterizzata da un tipo:

`int **`

Neanche questo tipo di puntatore contiene informazioni sul numero di righe e di colonne.

Definiamo dunque un nuovo **tipo**, costituito da una struttura contenente il puntatore **`int **`**, il numero di colonne ed il numero di righe.

```
typedef struct {
  int rows, cols;
  int ** data;
} t_matrix;
```

Libreria per la gestione/manipolazione delle matrici

```
/*
 * libreria funzioni su matrici di interi
 */

#include <stdio.h>
#include <stdlib.h>

typedef struct {
  int rows,cols;
  int ** data;
} t_matrix;

t_matrix new_matrix (int r, int c)
{
  t_matrix m;
  int i;

  m.rows = r;
  m.cols = c;
  m.data = (int **)malloc (r * sizeof (int *));
  for (i = 0;i < r;i++) {
    m.data [i] = (int *)malloc (c * sizeof (int));
  }
  return m;
}
```

Libreria per la gestione/manipolazione delle matrici

```
……

void delete_matrix (t_matrix m)
{
  int i;
  for (i = 0;i < m.rows;i++)
    free (m.data [i]);
  free (m.data);
}

void print_matrix (t_matrix m)
{
  int i,j;

  for (i = 0; i < m.rows; i++) {
    for (j = 0; j < m.cols; j++)
      printf ("%d ", m.data [i][j]);
    printf ("\n");
  }
}

……
```

Libreria per la gestione/manipolazione delle matrici

```
……

t_matrix transpose (t_matrix m)
{
  t_matrix t;
  int i,j;

  t = new_matrix (m.cols, m.rows);
  for (i = 0; i < m.rows; i++)
    for (j = 0; j < m.cols; j++)
      t.data [j][i] = m.data [i][j];
  return t;
}

……
```

Libreria per la gestione/manipolazione delle matrici

```
……
/* Main per il test delle funzioni */
int main (int argc, char * argv[])
{
  t_matrix mymatrix, mytranspose;
  int r,c, j, i;

  printf ("Inserisci il numero di righe: ");
  scanf ("%d", &r);
  printf ("Inserisci il numero di colonne: ");
  scanf ("%d", &c);
  mymatrix = new_matrix (r, c);

  for (i = 0; i < r; i++)
    for (j = 0;j < c;j++) {
    printf ("Inserisci l'elemento %d,%d:", i,j);
    scanf ("%d", &mymatrix.data [i][j]);
  }
  mytranspose = transpose (mymatrix);
  print_matrix (mytranspose);
  delete_matrix (mytranspose);
  delete_matrix (mymatrix);
}
```

“Homework”: minore di una matrice

Arricchiamo la libreria di gestione delle matrici con una funzione che determina il minore di una matrice, cioè la sottomatrice ottenuta cancellando la i-esima riga e la j-esima colonna:

```
t_matrix minor (t_matrix m, int i, int j);
```

Hint: non pensate a come si “cancella” una riga (o una colonna), ma eseguite una “copia” della matrice **m** “saltando” la i-esima riga e la j-esima colonna.