

# Matrice concatenata

---

Per rendere la rappresentazione di una matrice di numeri reali dinamica, in modo da poter cambiare le sue dimensioni dinamicamente, si decide di usare una struttura dati a puntatori.

Più precisamente, ogni elemento della struttura corrisponde ad un elemento della matrice, ed ha i seguenti attributi:

- `val` è il valore contenuto nell'elemento;
- `row` è la riga su cui l'elemento si trova, mentre `col` è la colonna;
- `right` è il riferimento all'elemento che si trova a destra, cioè sulla stessa riga, ma nella colonna successiva;
- `down` è il riferimento all'elemento che si trova in basso, cioè sulla stessa colonna, ma sulla riga successiva. Gli elementi sull'ultima colonna hanno `NULL` come valore dell'attributo `right`, mentre quelli sull'ultima riga hanno `NULL` come valore dell'attributo `down`.

Una matrice `M` è una struttura con un attributo, `head`, che è il riferimento all'elemento in posizione  $[1, 1]$  della matrice (`NULL` se la matrice è vuota).

Si veda l'esempio stampato.

Il programma in calce contiene un'implementazione parziale di tale struttura dati. Dovete:

1. completare la definizione dei tipi in maniera opportuna
2. scrivere una funzione `void stampaMatrice ( linkedMatrix m )` che stampa su standard input la matrice passata per argomento
3. scrivere una funzione `void stampaMatriceTrasposta ( linkedMatrix m )` che stampa su standard input la matrice trasposta di quella passata per argomento.

## Esempio di esecuzione 1

---

Eseguendo

```
./soluzione
```

avendo nel flusso di ingresso:

```
2 4
4      -8      2      1
-2      -9      0      6
```

il programma emette sul flusso di uscita:

```
4      -8      2      1
-2      -9      0      6

4      -2
-8      -9
2      0
1      6
```

## Esempio di esecuzione 2

Eseguendo

```
./soluzione
```

avendo nel flusso di ingresso:

```
4 2
4      2
8      9
2      0
1      6
```

il programma emette sul flusso di uscita:

```
4      2
8      9
2      0
1      6

4      8      2      1
2      9      0      6
```

## Programma da completare

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

```
struct element {
    ... val;
    ... row;
    ... col;
    ... right;
    ... Ódown;
};
```

```
typedef .... Element;
```

```
typedef struct {
    Element head;
    int rows;
    int cols;
} linkedMatrix;
```

```
linkedMatrix leggiMatrice( void );
linkedMatrix trasponiMatrice ( linkedMatrix m );
void stampaMatrice ( linkedMatrix m );
void stampaMatriceTrasposta ( linkedMatrix m );
```

```
int main () {
    linkedMatrix m = leggiMatrice();
    stampaMatrice(m);
    printf( "\n" );
    stampaMatriceTrasposta(m);
    return 0;
}
```

```
linkedMatrix leggiMatrice( void ){

    int r,c;
    scanf( "%d%d", &r,&c );
    int t = r*c;
    Element curr; // puntatore all'elemento corrente della matrice
    Element first = NULL; // puntatore al primo elemento della riga corrente
    Element top = NULL; // puntatore all'elemento sopra a quello corrente

    linkedMatrix m;
    curr = malloc( sizeof (struct element) );
```

```

m.head = curr;
m.rows = r;
m.cols = c;
first = curr;

int i=0, j=0, n;
while ( --t >= 0 ) {
    scanf( "%d", &n );
    curr -> val = n;
    curr -> row = i;
    curr -> col = j;
    curr -> down = NULL;

    // collego l'elemento top all'elemento corrente
    if( top ) {
        top -> down = curr;
    }

    if ( t == 0 ) {
        curr -> right = NULL;
        break;
    }

    // creo spazio per il prossimo elemento
    Element new = malloc( sizeof (struct element) );

    // posiziono new
    // a destra dell'elemento corrente oppure all'inizio della prossima riga
    // aggiornno first e top di conseguenza
    if ( j % c == c-1 ) {
        curr -> right = NULL;
        j = 0;

        first -> down = new;
        top = first;
        first = new;
        i++;
    }
    else {
        curr -> right = new;
        j++;
        if ( top )
            top = top->right;
    }

    // sposto curr sulla prossima posizione
    printElement( curr );
    curr = new;
}

```

```
}
```

```
return m;
```

```
}
```