

Matrici in memoria dinamica

I modo: allocazione di un vettore lineare di dimensione $r*c$

Esempio: matrice di interi con r righe e c colonne

```
int* m;  
m=new int [r*c];
```

Inizializzazione

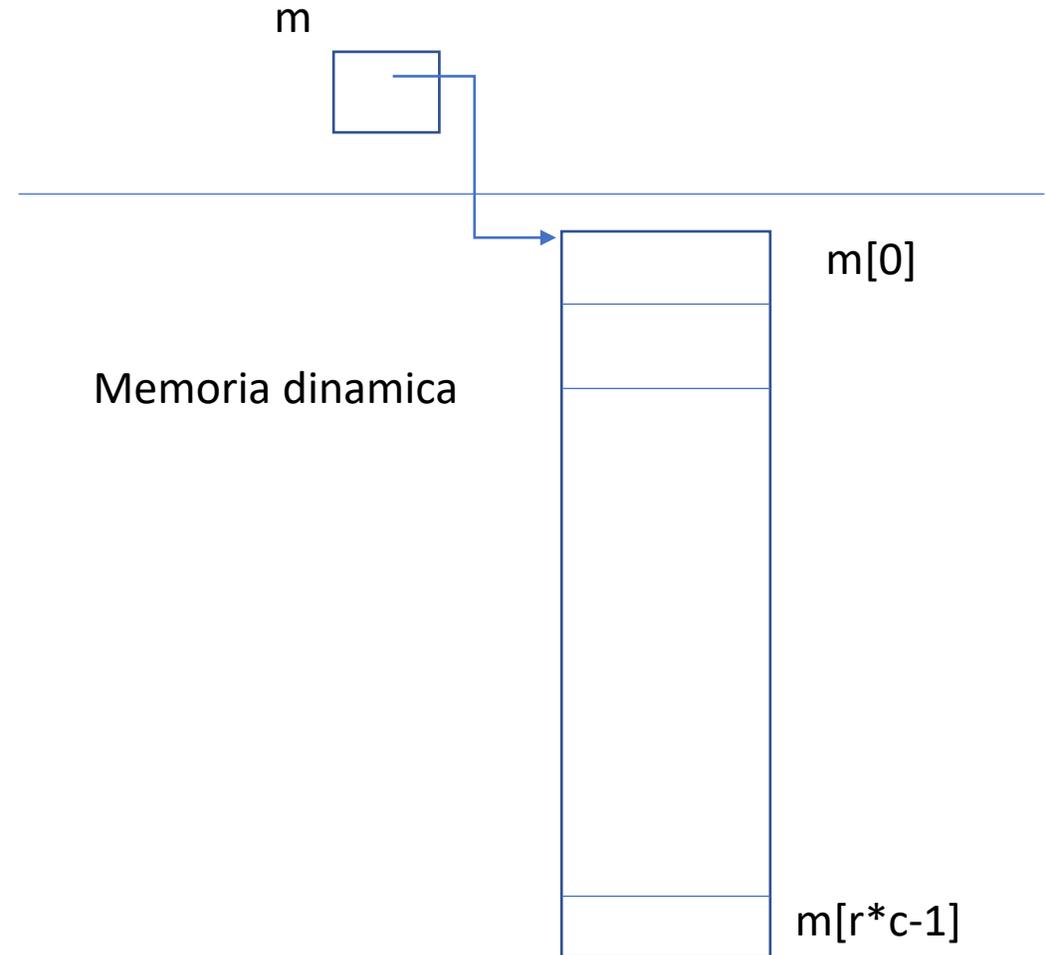
```
for (int i=0; i<r*c; i++)  
    cin >> m[i];
```

oppure

```
for (int i=0; i<r; i++)  
    for (int j=0; j<c; j++)  
        cin >> m[i*c +j];
```

```
Stampa della matrice
for (int i=0; i<r; i++){
    for (int j=0; i<c; i++){
        cout << m[i*c +j];
    }
    cout << '\n';
}
```

```
Deallocazione:
delete[]m;
```



Matrici in memoria dinamica

II modo: allocazione di un vettore di puntatori alle righe della matrice;
allocazione delle righe della matrice

Esempio: matrice di interi con r righe e c colonne

```
int** m; // puntatore a puntatore

m= new int*[r] // creazione del vettore di puntatori alle righe (r righe)

for (int i=0; i<r; i++)
    m[i]=new int[c]; // creazione delle righe: dimensione della riga
                    // numero di colonne c
```

Inizializzazione

```
for (int i=0; i<r; i++)  
    for (int j=0; i<c; i++)  
        cin >> m[i][j];    // accesso con indice
```

Deallocazione

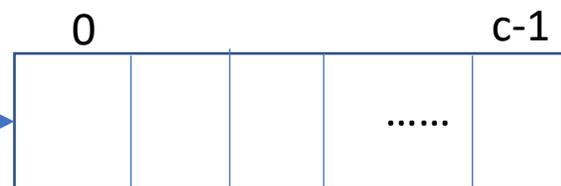
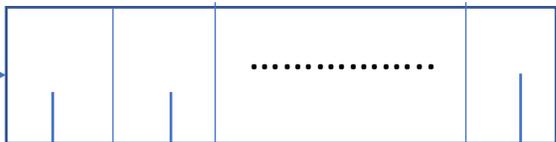
```
for (int i=0; i<r; i++)    // deallocazione delle righe  
    delete[]m[i]
```

```
delete[]m;    // deallocazione del vettore di puntatori
```

m



0 r-1



Memoria dinamica

