

L'array bidimensionale (detto anche **matrice**) è una *struttura di dati omogenea* (dati che sono **tutti dello stesso tipo**) definiti da un **nome collettivo** e da **2 dimensioni** dette **righe** e **colonne**.

La *dichiarazione* di una matrice in C/C++ è la seguente: **tipo nome [righe] [colonne];**

un elemento è individuato da **nome** e **posizione** di **riga** e di **colonna**: **nome [riga] [colonna]**

Le posizioni, dette **indici dell'array**, devono essere **valori interi**; la **posizione di riga** è un valore compreso tra **0** e **righe - 1**; la **posizione di colonna** è un valore compreso tra **0** e **colonne - 1**.

Il **numero di elementi** che la struttura contiene è dato dal **prodotto delle dimensioni**: **righe x colonne**

		indice di colonna						
		0	1	2	3	4	5	
i n d i c e	0	8	-.2	0	4	15	-7	< elementi della riga 0
	1	11	-5	0	-20	-30	25	
	2	3	8	6	-9	0	12	
	3	-10	-3	18	0	-12	16	

		indice di colonna						
		0	1	2	3	4	5	
i n d i c e	0	8	-.2	0	4	15	-7	
	1	11	-5	0	-20	-30	25	
	2	3	8	6	-9	0	12	
	3	-10	-3	18	0	-12	16	

^
elementi della colonna 4

La matrice di esempio è costituita da **4 righe** (dalla riga di posizione **0** alla riga di posizione **3 = 4 - 1**) e da **6 colonne** (dalla colonna di posiz. **0** alla colonna di pos. **5 = 6 - 1**); è costituita da **elementi di tipo intero**.

La *dichiarazione* per questa matrice è: **int matr [4] [6];**

Il **primo** elemento della **riga 0** è in colonna **0**, contiene il valore **8** e si indica con **matr [0] [0]**

L'**ultimo** elemento della **riga 0** è in colonna **5**, contiene il valore **-7** e si indica con **matr [0] [5]**

Il **primo** elemento della **colonna 4** è in riga **0**, contiene il valore **15** e si indica con **matr [0] [4]**

L'**ultimo** elemento della **colonna 4** è in riga **3**, contiene il valore **-12** e si indica con **matr [3] [4]**

La matrice contiene **24** elementi, infatti il prodotto delle dimensioni **4 x 6 = 24**.

Gli **indici** che individuano un particolare elemento della matrice possono essere:

- **costanti intere**; esempio i valori interi 2 e 5 **matr [2] [5] = 12;**
- valori contenuti in **variabili intere**; esempio k e h: **int k=1, h=4; matr [k] [h] = -30;**
- **formule** che restituiscono valori **interi**; es. se h vale 4, h-2 vale 2: **matr [3] [h-2] = 18;**

Le **matrici quadrate** hanno numero di **righe uguale** al numero di **colonne**, cioè dimensione orizzontale e verticale sono uguali; la dimensione **ORD** è detta **ordine** della matrice **tipo nome [ORD] [ORD]**;

Le matrici quadrate individuano 4 sottoinsiemi di elementi notevoli:

- gli elementi posti sulla **diagonale principale** e sulla **diagonale secondaria**

		indice di colonna			
		0	1	2	3
i n d i c e	0	1	2	3	4
	1	2	4	6	8
	2	3	6	9	12
	3	4	8	12	16

elementi diagonale principale

		indice di colonna			
		0	1	2	3
i n d i c e	0	1	2	3	4
	1	2	4	6	8
	2	3	6	9	12
	3	4	8	12	16

elementi diagonale secondaria

gli elementi della **diag. principale** hanno **posizione di riga = pos. di colonna**: **matr[k] [k]**

gli elementi della **diagonale secondaria** hanno posizioni indicate da: **matr[k] [ORD-1 - k]**
per questi elementi la somma degli indici **di riga** e **di colonna** vale **ORD-1**.

- gli elementi al di sopra o al di sotto della diagonale principale, individuano un **triangolo superiore** e un **triangolo inferiore**

		indice di colonna			
		0	1	2	3
i n d i c e	0	1	2	3	4
	1	2	4	6	8
	2	3	6	9	12
	3	4	8	12	16

elementi triangolo superiore

		indice di colonna			
		0	1	2	3
i n d i c e	0	1	2	3	4
	1	2	4	6	8
	2	3	6	9	12
	3	4	8	12	16

elementi triangolo inferiore

gli elementi del **triangolo superiore** hanno **indice di colonna maggiore dell' indice di riga**
matr[r] [c] con c > r

gli elementi del **triangolo inferiore** hanno **indice di colonna minore dell' indice di riga**
matr[r] [c] con c < r

CODIFICA in C/C++

```
#include <iostream>
using namespace std;
#define TAB '\t' // tabulazione
#define TRATT '-'
#define ACAPO '\n'
#define ACAPO2 "\n\n"
#define ORD 10
int main() {
    int r, c; //-- indici di riga e colonna
    int matr[ORD][ORD];
    // 2 cicli nidificati, qui non servono le parentesi graffe
    for (r = 0; r < ORD; r++) // ogni ciclo ha 1 istruzione
        for (c = 0; c < ORD; c++)
            matr[r][c] = (r+1) * (c+1);
    cout << ACAPO2;
    cout<<"---matrice quadrata di ordine 10---\n\n";
    for (r=0; r < ORD; r++) { // qui servono le parentesi
        // perché il ciclo esterno ha 2 istruzioni, il for e il cout
        for (c=0; c < ORD; c++)
            cout << matr[r][c] << TAB;
        cout << ACAPO2;
    } // per le diagonali basta un solo ciclo
    cout << "---diagonale principale---\n\n";
    for (int k=0; k < ORD; k++)
        cout << matr[k][k] << TAB;
    cout << ACAPO2;
    cout << "---diagonale secondaria---\n\n";
    for (int k=0; k < ORD; k++)
        cout << matr[k][ORD - 1 - k] << TAB;
    cout << ACAPO2;
    cout << "---triangolo superiore---\n\n";
    for (r=0; r < ORD; r++) {
        for (c=0; c < ORD; c++) // qui non servono le {}
            if (c > r) // il ciclo interno ha solo 1 istruz., if
                cout << matr[r][c] << TAB;
            else
                cout << TRATT << TAB;
        cout << ACAPO2;
    }
    cout << "---triangolo inferiore---\n\n";
    for (r=0; r < ORD; r++) {
        for (c=0; c < ORD; c++)
            if (c < r)
                cout << matr[r][c] << TAB;
            else
                cout << TRATT << TAB;
        cout << ACAPO2;
    }
}
```

A sinistra il programma che memorizza la **Tavola Pitagorica** (le 10 tabelline) in una **matrice quadrata di ordine 10** (parte della quale è stata rappresentata nella pagina precedente), la espone a video, scrive a video gli elementi della **diagonale principale** della **diagonale secondaria**, poi espone a video i **triangoli superiore e inferiore** indicando con un trattino (-) gli elementi della matrice che non fanno parte dei triangoli.

```
>>
>>matrPitagoraSupInfDiag.exe
---matrice quadrata di ordine 10---
1  2  3  4  5  6  7  8  9  10
2  4  6  8  10 12 14 16 18 20
3  6  9  12 15 18 21 24 27 30
4  8  12 16 20 24 28 32 36 40
5  10 15 20 25 30 35 40 45 50
6  12 18 24 30 36 42 48 54 60
7  14 21 28 35 42 49 56 63 70
8  16 24 32 40 48 56 64 72 80
9  18 27 36 45 54 63 72 81 90
10 20 30 40 50 60 70 80 90 100
---diagonale principale---
1  4  9  16  25  36  49  64  81  100
---diagonale secondaria---
10 18 24 28 30 30 28 24 18 10
---triangolo superiore---
-  2  3  4  5  6  7  8  9  10
-  -  6  8  10 12 14 16 18 20
-  -  -  12 15 18 21 24 27 30
-  -  -  -  20 24 28 32 36 40
-  -  -  -  -  30 35 40 45 50
-  -  -  -  -  -  42 48 54 60
-  -  -  -  -  -  -  56 63 70
-  -  -  -  -  -  -  -  72 80
-  -  -  -  -  -  -  -  -  90
-  -  -  -  -  -  -  -  -  -
---triangolo inferiore---
-  -  -  -  -  -  -  -  -  -
2  -  -  -  -  -  -  -  -  -
3  6  -  -  -  -  -  -  -  -
4  8  12 -  -  -  -  -  -  -
5  10 15 20 -  -  -  -  -  -
6  12 18 24 30 -  -  -  -  -
7  14 21 28 35 42 -  -  -  -
8  16 24 32 40 48 56 -  -  -
9  18 27 36 45 54 63 72 -  -
10 20 30 40 50 60 70 80 90 -
>>
```