

➤ **Matrici**

Array a più dimensioni.

Sintassi generale ::

tipo **nome** [dimensione 1][dimensione 2]...[dimensione n]

```
int a[10][15]
```

Lettura matrice N*M

```
-----  
for (i=0; i <N;i++)  
    for (j=0;j<M;j++)  
        cin >>Matrice[i] [j];  
-----
```

Produzione matrice N*M

```
-----  
for (i=0; i <N;i++)  
{  
    for (j=0;j<M;j++)  
        cout <<Matrice[i] [j];  
    cout <<endl;  
}  
-----
```

N=3 M=2

```
-----  
for (i=0; i <N;i++)  
    for (j=0;j<M;j++)  
        cin >>Matrice[i] [j];  
-----
```

i=0 i<N

j=0

Lettura di Matrice[0][0]

J=1 j<M Vero

Lettura di Matrice[0][1]

J=2 j<M Falso

i=1 i<N Vero

j=0

Lettura di Matrice[1][0]

J=1 j<M Vero

Lettura di Matrice[1][1]

J=2 j<M Falso

i=2 i<N Vero

j=0

Lettura di Matrice[2][0]

J=1 j<M Vero

Lettura di Matrice[2][1]

J=2 j<M Falso

i=3 i<N Falso



Esercizio

Assegnata una matrice di riempimenti N ed M,
determinare la somma dei suoi elementi

1	2	3	4	5
2	7	8	9	10
9	8	7	6	5

N = 3

M = 5

Somma = 86

```

#include <iostream>
using namespace std;
// Esercizio69 somma elementi di una matrice
int main()
{
    int matrice[10][10];
    int N,M;
    cout << "riempimento righe -->";cin >>N;cout <<endl;
    cout << "riempimento colonne -->";cin >>M;cout <<endl;
    int max,min,somma=0;
    for (int I=0;I<=N-1;I++)
        for (int J=0;J<=M-1;J++)
        {
            cout<<"Matrice["<<I<<"]["<<J<<"] =";
            cin >>matrice[I][J];
            somma=somma+matrice[I][J];
            cout<<endl;
        }
    cout <<"Somma =" << somma <<endl;
    system("PAUSE");
}

```

Esercizio 31

Assegnata una matrice di max 10*10 valori interi, determinare **valore max** e **valore min** e rispettive posizioni (*indici*)

```
#include <iostream.h>|
#include <stdlib.h>
using namespace std;
// Esercizio31 Ricerca max e min in matrice di interi
int main()
{
    int matrice[10][10];
    int N,M,I,J;
    cout << "riempimento righe -->";cin >>N; cout <<endl;
    cout << "riempimento colonne -->";cin >>M; cout <<endl;
    int max,min;
    int rigamax,rigamin=0, colonnamax,colonnamin=0;
```

```

for (I=0;I<=N-1;I++)
    for (J=0;J<=M-1;J++)
    {
        cin >>matrice[I][J];
        if ((I==0) && (J==0))
        {
            max=matrice[0][0];
            min=matrice[0][0];
        }
        if (matrice[I][J] >max)
        {
            max=matrice[I][J];
            rigamax=I;
            colonnamax=J;
        }
        if (matrice[I][J] <min)
        {
            min=matrice[I][J];
            rigamin=I;
            colonnamin=J;
        }
    }
}

```

```

    cout << "elemento max =" << max << "riga =" << rigamax+1
    << "colonna =" << colonnamax+1 << endl;
    cout << "elemento min =" << min << "riga =" << rigamin+1
    << "colonna =" << colonnamin+1 << endl;
    system("PAUSE");
}

```

riempimento righe -->4

riempimento colonne -->3

```

1  2  5  7  2  9  3  7  33  12  21  33
elemento max =33riga =3colonna =3
elemento min =1riga =1colonna =1
Premere un tasto per continuare . . .

```

Esercizio

Dato un array $A[N][M]$ di elementi interi, produrre:

Le somme di ogni riga

Le somme di ogni colonna

La riga di somma massima

La colonna di somma massima

N=4	1	2	3	4	Somme righe	10	23	9	22
M=4	2	6	7	8	Somme colonne	10	14	18	22
	3	1	2	3	Riga max	1			
	4	5	6	7	Colonna max	3			

```

#include <iostream.h>
#include <stdlib.h>
using namespace std;
int main()
{
    int A[10][10];
    int N,M,sommarighe[10],sommacolonne[10];
    int rigamax,colmax,posriga,poscol;
    cout<<"nrorighe =";cin>>N;
    cout<<"nro colonna =";cin>>M;
    // Lettura matrice
    for (int i=0;i<N;i++)
        for (int j=0;j<M;j++)
        {
            cout<<"A[  "<<i<<" , "<<j<<"  ]";cin>>A[i][j];
        }
    // Determina somma righe
    cout<<" Somma righe"<<endl;
    for (int i=0;i<N;i++)
    {
        sommarighe[i]=0;
        for (int j=0;j<M;j++)
            sommarighe[i]=sommarighe[i]+A[i][j];
        cout<<sommarighe[i]<<"  ";
    }
    cout<<endl;

```

```

// Determina somma colonna
cout<<" Somma colonna"<<endl;
for (int j=0;j<M;j++)
{
    sommacolonne[j]=0;
    for (int i=0;i<N;i++)
        sommacolonne[j]=sommacolonne[j]+A[i][j];
    cout<<sommacolonne[j]<<" ";
}
cout<<endl;
rigamax=sommarighe[0];
posriga=0;
for (int i=1 ;i<N;i++)
    if (rigamax <sommarighe[i])
    {
        rigamax=sommarighe[i];
        posriga=i;
    }
colmax=sommacolonne[0];
poscol=0;
for (int i=1 ;i<N;i++)
    if (colmax <sommacolonne[i])
    {
        colmax=sommacolonne[i];
        poscol=i;
    }
cout<<"rigamax ="<<posriga<<" colonnamax ="<<poscol<<endl;
system("PAUSE");
return 0;
}

```

```
nrighe =4
nro colonne =4
A[ 0,0 ] 11
A[ 0,1 ] 12
A[ 0,2 ] 13
A[ 0,3 ] 14
A[ 1,0 ] 15
A[ 1,1 ] 16
A[ 1,2 ] 17
A[ 1,3 ] 18
A[ 2,0 ] 19
A[ 2,1 ] 11
A[ 2,2 ] 12
A[ 2,3 ] 13
A[ 3,0 ] 14
A[ 3,1 ] 15
A[ 3,2 ] 16
A[ 3,3 ] 17
Somma righe
10 26 15 22
Somma colonne
19 14 18 22
rigamax =1  colonnamax =3
Premere un tasto per continuare...
```

Esercizio proposto

Dato un array $A[N][M]$ di elementi reali, produrre:

La riga di somma massima

La riga di somma minima

L'inversione delle due righe

La colonna di somma minima

L'inversione con la prima colonna

N=4	1	2	3	4
M=4	2	6	7	8
	3	1	2	3
	4	5	6	7

Riga somma max	2
Riga somma min	1
Colonna somma min	2

6	5	7	8
2	1	3	4
1	9	2	3
5	4	6	7

Esercizio

Assegnata una matrice quadrata di ordine N prefissato, calcolare la somma degli elementi della diagonale principale e quella degli elementi della diagonale secondaria

L'algoritmo deve avere validità sia per matrici d'ordine pari che dispari

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4

N = 5

El. Diag. Principale → 0,0 1,1 2,2 3,3 4,4 j=i
 El. Diag.Secondaria → 0,4 1,3 2,2 3,2 4,0 j=N - i -1

i=0	mat[0][0]	mat[0][4]
i=1	mat[1][1]	mat[1][3]
i=2	mat[2][2]	mat[2][2]
i=3	mat[3][3]	mat[3][1]
i=4	mat[4][4]	mat[4][0]

Sommap=Sommap+mat [i][i]
 Sommad=Sommad+mat [i][N - i -1]

```
#include <iostream>
using namespace std;
// Assegnata una matrice quadrata di ordine N
// determinare la somma degli elementi sulla
// diagonale principale e la somma di quelli
// sulla digonale secondaria
int main()
{
    float mat[10][10];
    int N, j ;
    float sommap=0.0, sommad=0.0;
    cout<<"Riempimento =";cin>>N;
// Lettura matrice
    for (int i=0; i<N; i++)
        for (int j=0; j<N; j++)
        {
            cout<<"elemento "<<i<<","<<j<<" =";
            cin>>mat[ i ][ j ];
        }
}
```

```
// Determina somma delle diagonali
```

```
for (int i=0; i<N; i++)  
{  
    sommap=sommap+mat[ i ][ i];  
    sommad=sommad+mat[ i ][ N-i-1];  
}
```

```
// Risultati
```

```
cout<<"somma diag. principale ="<<sommap<<endl;  
cout<<"somma diag. secondaria ="<<sommad<<endl;  
system("PAUSE");  
return EXIT_SUCCESS;  
}
```

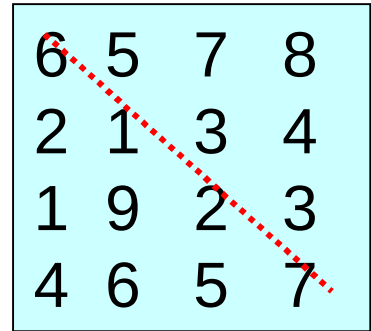
Esercizio proposto

Dato un array $A[N][N]$ di elementi interi, determinare:

La somma degli elementi della diagonale principale
L'elemento massimo e posizione fra gli elementi della diagonale principale

$N=4$

Somma	16
Elem max	7
Pos max	4,4



6	5	7	8
2	1	3	4
1	9	2	3
4	6	5	7

Esercizio 135

Assegnata in ingresso una matrice quadrata A d'ordine N , si generi la matrice trasposta

matrice input

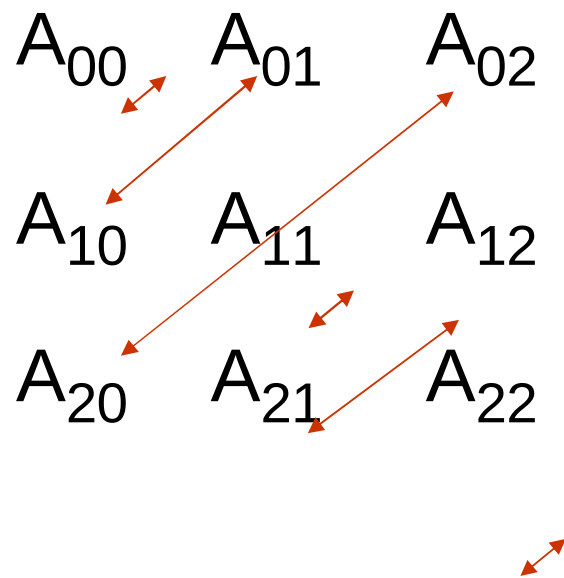
1	2	3
2	5	6
7	8	9

matrice trasposta

1	4	7
2	5	8
3	6	9

Metodo di soluzione

Scambio degli elementi $A[i][j]$ con gli elementi $A[j][i]$
(viene scambiata ogni riga d'ordine N con la colonna di pari ordine)



$i \ j$	$j \ i$
A_{00}	A_{00}
A_{01}	A_{10}
A_{02}	A_{20}
A_{11}	A_{11}
A_{12}	A_{21}
A_{22}	A_{22}

i da 0 a $N-1$

j da i a $N-1$

```

#include <iostream>
using namespace std;
// Esercizio135  Generazione matrice trasposta
// di matrice quadrata
int main()
{
    int A[10][10];
    int N, temp;
    cout<< "Riempimento=";cin >>N;cout <<endl;
    for (int i=0; i<N; i++)
    {
        cout <<"Riga " <<i <<" = ";
        for (int j=0; j<N; j++)
            cin >>A[ i ][ j ];
    }
    for (int i=0;i<N-1;i++)
    for (int j=i;j<N;j++)
    {
        temp=A[ i ][ j ];
        A[ i ][ j ]=A[ j ][ i ];
        A[ j ][ i ]=temp;
    }
}

```

```
cout<<"MATRICE TRASPOSTA" <<endl;
for (int i=0;i<N;i++)
{
    for (int j=0; j<N; j++)
        cout <<A[ i ][ j ] <<" ";
    cout <<endl;
}
system("PAUSE");
return EXIT_SUCCESS;
}
```

```
Riempimento=3
Riga 0 = 1 2 3
Riga 1 = 4 5 6
Riga 2 = 7 8 9
MATRICE TRASPOSTA
1 4 7
2 5 8
3 6 9
Premere un tasto per continuare . . .
```

Esercizio 124

Assegnata una matrice **A** di interi e di riempimenti **RIEMP1** e **RIEMP2** si generino da essa due vettori **VMAX** e **VMIN** contenenti rispettivamente i valori massimi di ogni riga e i minimi di ogni colonna

1	12	-7	6	5
1	9	8	4	3
1	22	23	25	4
7	44	12	2	6
2	3	26	18	9

VMAX	12	9	25	44	26
VMIN	1	3	-7	2	3

Ipotesi algoritmo

Leggi Riempimenti e matrice A

Per ogni riga della matrice

for (i=0;<Riemp1;i++)

blocco1

Assumi primo elemento riga come max corrente **max=A[i] [1]**

Per indice corrente di colonna da 1 a <Riemp2

blocco2

for (j=1;<Riemp2;j++)

se max corrente <elemento corrente matrice

aggiorna max corrente con elemento corrente

fine blocco2

If (max <A[i] [j]) max= A[i] [j]

memorizza max corrente in VMAX

fine blocco1

VMAX[i] = max

```
#include <iostream>
using namespace std;
int main()
{
    int A[10][10];
    int VMAX[10],VMIN[10];
    int Riemp1,Riemp2;
    int max,min;
    cout << "Riempimento1 -->"; cin>>Riemp1;cout <<endl;
    cout << "Riempimento1 -->"; cin>>Riemp2;cout <<endl;
    // Lettura matrice A
    for (int i=0; i<Riemp1; i++)
        for (int j=0; j<Riemp2; j++)
        {
            cout << "A[ " <<i << ", " <<j << "]=";cin >>A[ i ][ j ];
            cout <<endl;
        }
}
```

```
// Generazione VMAX
```

```
for (int i=0; i<Riemp1; i++)  
{  
    max=A[ i ][ 0 ];  
    for (int j=1; j<Riemp2; j++)  
        if (max <A[ i ][ j ])  
            max=A[ i ][ j ];  
    VMAX[ i ]=max;  
}
```

```
// Generazione VMIN
```

```
for (int j=0; j<Riemp2; j++)  
{  
    min=A[0][j];  
    for (int i=1; i<Riemp1; i++)  
        if (min >A[ i ][ j ])  
            min=A[ i ][ j ];  
    VMIN[ j ]=min;  
}
```

```
// Stampe
```

```
cout << " VMAX " <<endl;  
for (int i=0; i<Riemp1; i++)  
    cout <<VMAX[ i ]<< " ";  
cout <<endl;  
cout << " VMIN " <<endl;  
for (int j=0; j<Riemp2; j++)  
    cout <<VMIN[ j ]<<" ";  
cout <<endl;  
system("PAUSE");  
return EXIT_SUCCESS;  
}
```

Esercizio 106

Assegnata una matrice di interi di riempimenti N ed M,
eliminare la riga in posizione K

N=4 M=5 K=2

	1	2	3	4	5		1	2	3	4	5
	2	7	8	9	0		2	7	8	9	0
→	3	8	3	1	5	→	6	0	2	9	4
└	6	0	2	9	4						

```

#include <iostream>
using namespace std;
// Esercizio106- Eliminazione riga da matrice
int main()
{
    int matrix[10][10];
    int N,M,k;
    cout <<"nro righe =";cin>>N;cout<<endl;
    cout <<"nro colonne =";cin>>M;cout<<endl;
// Lettura matrice per righe
    for ( int i=0; i<N; i++)
    {
        cout <<"riga " <<i<<" = ";
        for (int j=0; j<M; j++)
            cin >>matrix[ i ][ j ];
        cout<<endl;
    }
    cout <<"riga da eliminare ";cin >>k;cout<<endl;

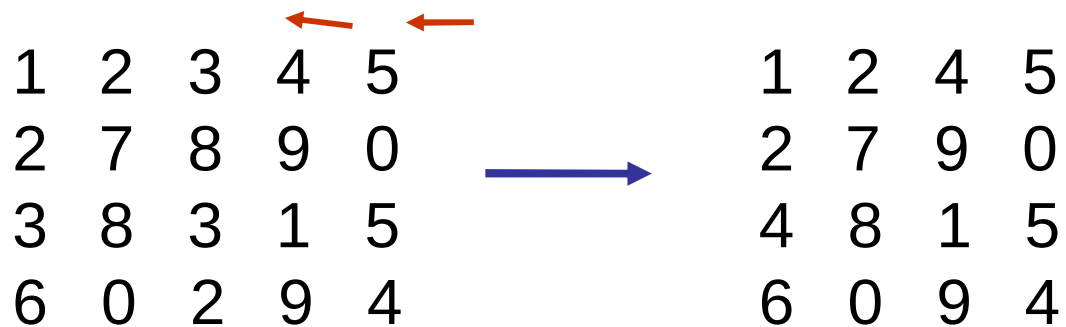
```

`if (k == (N-1))` ← Caso k sia ultima riga
 `N=N-1;`
`else`
 `for (int i=k; i<N-1; i++)`
 `for (int j=0; j<M; j++)`
 `matrix[ i ][ j ] =matrix[ i+1 ][ j ];`
`for (int i=0; i<N-1; i++)`
 {
 `for (int j=0; j<M; j++)`
 `cout<<matrix[ i ][ j ]<<" ";`
 `cout<<endl;`
 }
 `system("PAUSE");`
 `return 0;`
}

Esercizio 105

Assegnata una matrice di interi di riempimenti N ed M, eliminare la colonna in posizione K

N=4 M=5 K=2



```

#include <iostream>
using namespace std;
// Esercizio105- Eliminazione colonna da matrice
int main()
{
    int matrix[ 10 ][ 10 ];
    int N,M,k;
    cout <<"nro righe =">>N;cout<<endl;
    cout <<"nro colonne =">>M;cout<<endl;
// Lettura matrice per righe
    for (int i=0; i<N; i++)
    {
        cout <<"riga "<<i<<" = ";
        for (int j=0; j<M; j++)
            cin >>matrix[ i ][ j ];
        cout<<endl;
    }
    cout <<"colonna da eliminare ">>k;cout<<endl;

```

```

    if (k == (M-1))
// Se ultima colonna
        cout<<"colonna eliminata "<<k;
    else
// Non ultima colonna
        for (int j=k;j<M-1;j++)
            for (int i=0; i<N; i++)
                matrix[ i ][ j ] =matrix[ i ][ j+1 ];
    for (int i=0; i<N; i++)
    {
        for (int j=0; j<M-1; j++)
            cout<<matrix[ i ][ j]<<" ";
        cout<<endl;
    }
    system("PAUSE");
    return 0;
}

```

Esercizio 137

Data una matrice di interi A ($N * M$) determinare gli elementi che risultano ***contemporaneamente*** massimo della riga e della colonna di appartenenza

8	17	12	5
1	12	44	6
15	4	9	2
3	5	2	1

1	0	1
2	1	2
15	2	0

Ipotesi algoritmo

Per ogni riga

Valuta elemento max e colonna di appartenenza

Se è anche il max di tale colonna

Produci stampa

```
#include <iostream>
using namespace std;
// Esercizio137 Determinazione elementi max di riga e colonna
int main()
{
    int A[10][10];
    bool flag;
    int pos,max,N,M;
    cout << "Nro righe =";cin >>N;cout <<endl;
    cout << "Numero colonne =";cin >>M; cout <<endl;
    for (int i=0;i<N;i++)
        for (int j=0; j<M; j++)
            cin >> A[ i ][ j ];
    for (int i=0; i<N; i++)
    {
        max=A[ i ][ 0 ];
        pos=0;
```

```

// Determina max di riga e posizione
for (int j=1; j<M; j++)
    if (max < A[i][j])
    {
        max=A[i][j];  pos=j;
    }
// Determina se anche max di colonna
flag=true;
for (int K=1; K<N; K++)
    if (A[K][pos] >max)
        flag=false;
if (flag )
    cout <<"elem="<<max <<" riga "<<i <<" col "<<pos <<endl;
}
system("PAUSE");
return EXIT_SUCCESS;
}

```

Esercizio 122

Assegnate due matrici di interi A e B , determinare la matrice prodotto C

Condizione necessaria per il prodotto di due matrici:

$\text{Nr colonne_primamatrice} = \text{Nr righe_secondamatrice}$

$A[m][n]$ $B[n][k]$ prodotto possibile

$A[m][n]$ $B[m][k]$ prodotto impossibile

Prodotto di due matrici $A[m][n] * B[n][k]$ è una terza matrice $C[m][k]$ con il generico elemento

$$C_{i,j} = \sum_{k=1}^n A_{i,k} * B_{k,j}$$

$i \rightarrow 1, m$
 $j \rightarrow 1, l$

1	2	3
4	5	6

A(2,3)
m,n

1	2
3	4
5	6

B(3,2)
n, l

22	28
49	64

C(2,2)
m,l

per $m=2$
 $n=3$
 $l=2$

$$C_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$$

$$C_{12} = a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32}$$

$$C_{21} = a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31}$$

$$C_{22} = a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32}$$

```
for (int i=0; i<m; i++)
  for (int j=0; j<l; j++)
  {
    C[i][j]=0;
    for (int k=0; k<n; k++)
      C[i][j]=C[i][j]+A[i][k]+
        B[k][j];
```

```

#include <iostream>
using namespace std;
// Esercizio122 Prodotto matrici
int main()
{
    int A[10][10], B[10][10], C[10][10];
    int M1,N1,M2,N2;
    cout << "Numero righe matrice A=";cin >>M1 ;cout <<endl;
    cout << "Numero colonne matrice A=";cin >>N1;cout <<endl;
    cout << "Numero righe matrice B=";cin >>M2;cout <<endl;
    cout << "Numero colonne matrice B=";cin >> N2;cout <<endl;
    if (N1 == M2)
    {
        // Lettura matrice A
        for (int i=0;i<M1;i++)
            for (int j=0;j<N1;j++)
            {
                cout << "A[ " <<i <<"," <<j << " ] =";cin >>A[i][j];cout<<endl;
            }
        // Lettura matrice B
        for (int i=0;i<M2;i++)
            for (int j=0;j<N2;j++)
            {
                cout <<"B[ " <<i <<"," <<j <<" ] =";cin >>B[i][j];cout<<endl;
            }
    }
}

```

```

// Produzione matrice prodotto C
    for (int i=0;i<M1;i++)
        for (int j=0;j<N2;j++)
        {
            C[i][j]=0;
            for (int k=0;k<N1;k++)
                C[i][j]=C[i][j]+A[i][k]*B[k][j] ;
        }
// Stampa matrice C
    cout << "Matrice Prodotto" <<endl;
    for (int i=0;i<M1;i++)
    {
        cout <<endl;
        for (int j=0;j<N2;j++)
            cout <<C[i][j] <<" ";
        }
    cout <<endl;
}
else
    cout <<"Prodotto non possibile" <<endl;
system("PAUSE");
return EXIT_SUCCESS;
}

```