

Calcolo del punto di sella

1. Introduzione e requisiti del problema
2. Specifica
3. Progetto della soluzione
4. Codifica

1. Introduzione e requisiti del problema

Requisiti del problema

Scrivere una funzione C++ che, operando su di una matrice di numeri interi distinti passata come parametro, restituisce tre parametri interi che sono relativi all'eventuale punto di sella della matrice:

- riga del punto di sella;
- colonna del punto di sella;
- parametro intero di valore 0 se la matrice non contiene punti di sella, oppure di valore 1 se la matrice contiene un punto di sella.

In questa esercitazione sarà illustrata la modalità di lettura di un *array bidimensionale*.

Tale struttura dati consente di rappresentare le matrici. In particolare il problema posto richiede di calcolare il *punto di sella* di una matrice

Definizione

Il punto di sella di una matrice è:

l'elemento che è il minimo della riga

e

il massimo della colonna

Nel caso la matrice sia composta da elementi distinti il punto di sella se esiste è unico.

Esempio

33	7	71
19	12	83
29	4	15
39	2	91

Il **punto di sella** esiste ed è il valore **12**
(seconda riga, seconda colonna)
perché è il valore minimo della seconda riga
(i cui elementi sono 19,12,83)
ed è il valore massimo della seconda colonna
(i cui valori sono 7,12,4,2)

Casi di test

caso 1

Il punto di sella esiste

Input:

33	7	71
19	12	83
29	4	15
39	2	91

33	7	71
19	12	83
29	4	15
39	2	91

Output:

Punto di sella=12
riga=2
colonna=2

caso 2

Il punto di sella non esiste

Input:

5	17	9
91	4	19
82	12	7
6	20	33

Output:

Punto di sella=non esiste

Il problema fondamentale è la scansione di una matrice secondo un criterio particolare.

```
int arr[3];
```

2	3	4
34	0	5

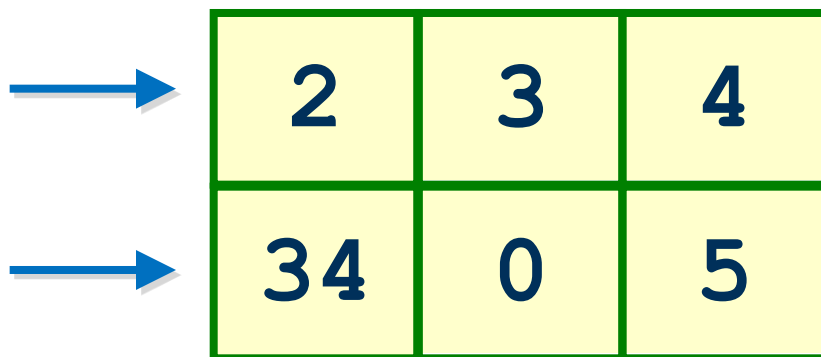
Il problema fondamentale è la scansione di una matrice secondo un criterio particolare.

```
int matrice[2][3];
```

2	3	4
34	0	5

Il problema fondamentale è la scansione di una matrice secondo un criterio particolare.

```
int matrice[2][3];
```



2	3	4
34	0	5

Il problema fondamentale è la scansione di una matrice secondo un criterio particolare.

```
int matrice[2][3];
```

2	3	4
34	0	5



Il problema fondamentale è la scansione di una matrice secondo un criterio particolare.

```
int matrice[2][3];
```

2	3	4
34	0	5

Problema: lettura della matrice

Problema: *lettura della matrice*

```
int ar1[MAXDIM];  
int i;  
...  
...  
for(i=0;i<MAXDIM;i++)  
{  
...  
cout<<ar1[i]<<endl;  
...  
}
```

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];  
int i,j;  
...  
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe  
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della  
        // i-esima riga  
        cout<<matr1[i][j];  
...
```

2	4	5
0	-6	76
3	5	87

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];  
int i,j;  
...  
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe  
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della  
        // i-esima riga  
        cout<<matr1[i][j];  
...
```

2	4	5
0	-6	76
3	5	87

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];  
int i,j;  
...  
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe  
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della  
        // i-esima riga  
        cout<<matr1[i][j];  
...
```

2	4	5
0	-6	76
3	5	87

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

i=0

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];  
int i,j;  
...  
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe  
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della  
        // i-esima riga  
        cout<<matr1[i][j];  
...
```

2	4	5
0	-6	76
3	5	87

$i=0$
 $j=0$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

$i=0$
 $j=1$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

$i=0$
 $j=2$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];  
int i,j;  
...  
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe  
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della  
        // i-esima riga  
        cout<<matr1[i][j];  
...
```

2	4	5
0	-6	76
3	5	87

$i=1$
 $j=2$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

$i=1$
 $j=0$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

$i=1$
 $j=1$

Problema: *lettura della matrice*

```
int matr1[RIGHE][COLONNE];
int i,j;
...
for(i=0;i<RIGHE;i++)//ciclo esterno che legge le righe
    for(j=0;j<COLONNE;j++)//ciclo interno che legge le colonne della
        // i-esima riga
        cout<<matr1[i][j];
...
```

2	4	5
0	-6	76
3	5	87

$i=1$
 $j=2$

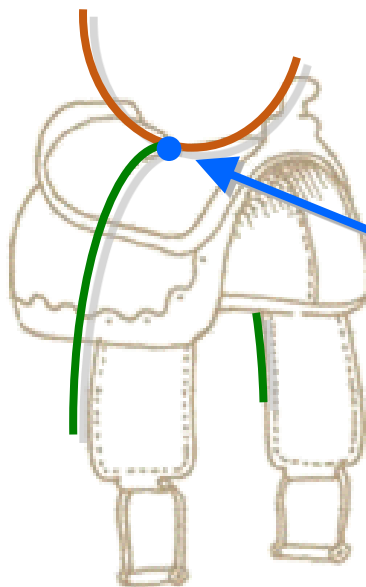
Creazione della funzione

Dati di input:

matrice

Dati di output:

elemento della radice,
coordinate della matrice



Sella

elemento minimo nella riga,
massimo nella colonna

Algoritmo di base: base

leggere riga per riga e controllare
se l'elemento minimo della riga è anche
il massimo della colonna

Pseudocodice

```
esiste=FALSO;  
  
do  
  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;
```

```
do
```

```
{
```

```
    Determina la colonna del  
    minimo della riga corrente
```

```
    Determina la riga del massimo  
    in tale colonna
```

```
    if(riga corrente==riga del massimo)
```

```
    {
```

```
        esiste=VERO;
```

```
    }
```

```
}while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;  
  
do  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;  
  
do  
  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;  
  
do  
  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;  
  
do  
  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;  
  
do  
  
{  
    Determina la colonna del  
    minimo della riga corrente  
  
    Determina la riga del massimo  
    in tale colonna  
  
    if(riga corrente==riga del massimo)  
    {  
        esiste=VERO;  
    }  
}  
  
}while((!esiste)&&(riga corrente<righe))
```

Pseudocodice

```
esiste=FALSO;
```

```
do
```

```
{
```

```
    Determina la colonna del  
    minimo della riga corrente
```

```
    Determina la riga del massimo  
    in tale colonna
```

```
    if(riga corrente==riga del massimo)
```

```
    {
```

```
        esiste=VERO;
```

```
    }
```

```
}while((!esiste)&&(riga corrente<righe))
```


Esempio

7	4	5
6	0	76
3	1	87

Esempio

7	4	5
6	0	76
3	1	87

riga corrente=0
colonna corrente=0
colonna minima=0
valore corrente=7
valore minimo=7

Esempio

7	4	5
6	0	76
3	1	87

riga corrente=0
colonna corrente=1
colonna minima=0
valore corrente=4
valore minimo=7

Esempio

7	4	5
6	0	76
3	1	87

riga corrente=0
colonna corrente=1
colonna minima=1
valore corrente=4
valore minimo=4

Esempio

7	4	5
6	0	76
3	1	87

riga corrente=0
colonna corrente=2
colonna minima=1
valore corrente=5
valore minimo=4

Esempio

7	4	5
6	0	76
3	1	87

Esempio

7	4	5
6	0	76
3	1	87

Punto di sella