

`int CellaAdiacente (int Ascissa, int Ordinata, int NumRighe, int NumColonne, list<int> MatricePosizionamento, int *Orientamento, int Id)`

Valori validi:

Ascissa: compresa tra 0 e NumRighe

Ordinata: compresa tra 0 e NumColonne

MatricePosizionamento: lista di NumRighe * NumColonne interi

NumRighe $\geq 10^k$ ($k \in \mathbb{N}$, $k > 0$)

NumColonne $\geq 10^k$ ($k \in \mathbb{N}$, $k > 0$)

Orientamento: 1, 2, 3

Id > 0

Input comune:

MatricePosizionamento:

10 10 0 0

14 14 11 0

8 8 0 0

NumRighe = 3

NumColonne = 4

Input:

Orientamento = 1

Id = 10

Ascissa = 2

Ordinata = 0

Output ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 1
Id = 10
Ascissa = 2
Ordinata = 0
Restituisce: 1
```

Output atteso:

Restituisce 1

Input:

Orientamento = 1

Id = 10

Ascissa = 3

Ordinata = 0

Output ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 1
Id = 10
Ascissa = 3
Ordinata = 0
Restituisce: 0
```

Output atteso:

Restituisce 0

Input:

Orientamento = 2

Id = 11

Ascissa = 2

Ordinata = 0

Output ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 2
Id = 11
Ascissa = 2
Ordinata = 0
Restituisce: 1
```

Output atteso:

Restituisce 1

Input:

Orientamento = 2

Id = 11

Ascissa = 3

Ordinata = 1

Output ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 2
Id = 11
Ascissa = 3
Ordinata = 1
Restituisce: 0
```

Output atteso:

Restituisce 0

Input:

Orientamento = 3

Id = 11

Ascissa = 3

Ordinata = 1

Output ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 3
Id = 11
Ascissa = 3
Ordinata = 1
Restituisce: 1
```

Output atteso:

Restituisce 1

Input:

Id < 0

Output atteso:

Restituisce 0 (indipendentemente dagli altri valori)

Output Ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 2
Id = -1
Ascissa = 3
Ordinata = 1
Restituisce: 0
```

Input:

Orientamento ≠ 1, 2, 3

Output atteso:

Restituisce 0 (indipendentemente dagli altri valori)

Output Ottenuto:

```
MatricePosizionamento:
10 10 0 0
14 14 11 0
8 8 0 0
NumRighe = 3
NumColonne = 4
Orientamento = 0
Id = 10
Ascissa = 3
Ordinata = 1
Restituisce: 0
```