

`int CellaValidaPosizionamento ( int Ascissa, int Ordinata, int LunghezzaNave, giocatore *Utente, torneo *Torneo)`

Valori validi:

Ascissa: compresa tra 0 e NumRighe delle Impostazioni del torneo

Ordinata: compresa tra 0 e NumColonne delle Impostazioni del torneo

Lunghezza Nave: 2,3,4,5

Giocatore: Giocatore con tutti i suoi campi settati correttamente

Torneo: Torneo con tutti i suoi campi settati correttamente

Input comune:

MatricePosizionamento:

10 10 0 0

14 14 11 0

8 8 0 0

NumRighe = 3

NumColonne = 4

Input :

LunghezzaNave = 3

Ascissa = 3

Ordinata = 1

Output atteso:

Restituisce 2

Output ottenuto:

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

Input :

LunghezzaNave = 2

Ascissa = 3

Ordinata = 1

Output ottenuto:

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

Output atteso:

Restituisce 3

Input :

LunghezzaNave = 2

Ascissa = 2

Ordinata = 0

Output ottenuto:

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

Output atteso:

Restituisce 1

Input :

Giocatore e/o Torneo non validi

Output atteso:

Crash della funzione

Input :

Ordinata \* NumColonne + Ascissa > | Giocatore.MatricePosizionamento |

Output atteso:

Crash della funzione

Input :

LunghezzaNave > 5

Output atteso:

Restituisce un valore diverso (0,1,2,3) in base alla presenza (o assenza) di spazio sufficiente a posizionare la nave più lunga di 5 celle