
FONDAMENTI DI INFORMATICA – INGEGNERIA BIOMEDICA

Una `BattagliaNavale` è identificata da un nome che consiste in una stringa di lettere minuscole o maiuscole dell'alfabeto (con lunghezza minore o uguale a 20) e da una scacchiera di posizioni su cui vengono posizionate navi di dimensione unitaria. Una nave può essere affondata/non affondata. Le righe della scacchiera sono indicate da numeri interi a partire da 1: 1 è la prima riga, 2 è la seconda riga e così via; analogamente, le colonne sono indicate con numeri interi a partire da 1. Implementare le seguenti operazioni che possono essere effettuate su una `BattagliaNavale`:

✓ **`InizializzaBattagliaNavale (b, s, n, p)`**

Funzione che inizializza una `BattagliaNavale` `b` con nome indicato dalla stringa `s`, `n` righe e `p` colonne. Inizialmente non ci sono navi sulla scacchiera.

✓ **`posizionaNave (b, r, c)`**

Funzione che posiziona una nave nella colonna `c` della riga `r`. Se la posizione è già occupata, la funzione lascia la battaglia navale inalterata e restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.

✓ **`colpisciNave (b, r, c)`**

Funzione che colpisce la colonna `c` della riga `r`. Se la posizione è occupata da una nave non ancora affondata, la nave diventa affondata e la funzione restituisce `true`; altrimenti restituisce `false`.

✓ **`stampaBattagliaNavale (b)`**

Funzione che stampa la `BattagliaNavale` `b`. Viene stampato l'identificatore della battaglia navale e, per ogni posizione viene visualizzato se la posizione è libera, occupata da una nave affondata oppure occupata da una nave non affondata secondo il formato seguente:

Nome della battaglia navale: terza partita

```

1234567
1   x
2   o
3  oxx
```

In questo esempio, il nome della battaglia navale è "terza partita", la battaglia navale ha 3 righe e 7 colonne. Nella prima riga c'è una nave affondata nella colonna 4; nella seconda riga c'è una nave non affondata nella colonna 2, nella terza riga c'è una nave non affondata nella colonna 3 e ci sono due navi affondate: una nella colonna 4 e una nella colonna 5.

✓ **`aggiungiNavi (b, k)`**

Funzione che posiziona `k` navi nelle prime `k` posizioni libere adiacenti sulla stessa riga, a partire dalla prima riga e dalla prima colonna. Se l'operazione fallisce, l'operatore lascia la battaglia navale inalterata.

Mediante il Linguaggio C++, realizzare il tipo `BattagliaNavale` definito dalle precedenti specifiche utilizzando le strutture. Implementare il tipo di dato con una matrice. Gestire le eventuali situazioni di errore.

Esempio di funzione main()

// Il contenuto dello stream d'output standard è rappresentato racchiuso tra i simboli di commento /**/

```
int main(){
    BattagliaNavale b;
    InizializzaBattagliaNavale (b, "prima partita", 3, 7);
    stampaBattagliaNavale(b);
    /*
        Nome della battaglia navale: prima partita

        1234567
        1
        2
        3
    */

    posizionaNave (b, 2, 1);
    posizionaNave (b, 9, 4);
    posizionaNave (b, 1, 6);
    stampaBattagliaNavale(b);
    /*
        Nome della battaglia navale: prima partita

        1234567
        1      o
        2 o
        3
    */

    colpisciNave (b, 1, 6);
    colpisciNave (b, 3, 7);
    stampaBattagliaNavale(b);
    /*
        Nome della battaglia navale: prima partita

        1234567
        1      x
        2 o
        3
    */

    aggiungiNavi(b, 3);
    aggiungiNavi(b, 7);
    stampaBattagliaNavale(b);
    /*
        Nome della battaglia navale: prima partita

        1234567
        1 ooo  x
        2 o
        3 ooooooo
    */

    return 0;
}
```