

FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

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 (numero minore o uguale a 26) sono indicate da lettere maiuscole dell'alfabeto: A è la prima riga, B è la seconda riga e così via; le colonne (numero minore o uguale a 9) sono indicate con numeri interi a partire da 1. Implementare le seguenti operazioni che possono essere effettuate su una `BattagliaNavale`:

- ✓ **`BattagliaNavale b(n, p, s)`**
Costruttore che inizializza una `BattagliaNavale b` con nome indicato dalla stringa `s`, `n` righe e `p` colonne. Inizialmente non ci sono navi sulla scacchiera.
- ✓ **`b.posiziona(r, c)`**
Funzione che posiziona una nave nella colonna `c` della riga `r`, dove `c` è un intero e `r` è un carattere. 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`.
- ✓ **`b.colpisci(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`.
- ✓ **`cout << b`**
Operatore di uscita per il tipo `BattagliaNavale`. 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
A   x
B   o
C   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.

- ✓ **`!b`**
Operatore di negazione logica che restituisce il numero delle navi non ancora affondate.
- ✓ **`b1=b2`**
Operatore di assegnamento.
- ✓ **`b+=k`**
Operatore di somma e assegnamento 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.
- ✓ **`~BattagliaNavale()`**
Distruttore.

Utilizzando il linguaggio C++, realizzare il tipo di dato astratto `Battaglia Navale` definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA

AVVIO E IDENTIFICAZIONE

- Avviare la macchina in modalità diskless, scegliere “Fondamenti di Informatica I” ed effettuare il login:
 nome: studenti
 password: studenti
- Aprire un terminale e spostarsi sulla cartella ‘elaborato’ (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.
- Dare il comando `$ ident`, sempre da dentro la cartella. Lo script richiede i propri dati (cognome, nome, numero di matricola e password (la password **non va dimenticata** in quanto è indispensabile per scaricare da internet il proprio elaborato a consegna avvenuta). Il comando `ident` crea il file `matricola.txt` nella cartella corrente. Lo script può essere lanciato più volte, in tal caso il file `matricola.txt` viene sovrascritto. Per verificare che il file sia stato creato e che il contenuto sia quello giusto dare il comando (la password è codificata):
`$ cat /home/studenti/elaborato/matricola.txt`
- A questo punto il docente verifica che tutti gli studenti abbiano effettuato l’identificazione, dopodichè provvede a inviare i seguenti file nella cartella `elaborato` del proprio PC:
 - ✓ `compito.h`
 - ✓ `compito.cpp`
 - ✓ `main.cpp`

Controllare pertanto che questi file, insieme al file `matricola.txt`, siano presenti sul proprio elaboratore.

SVOLGIMENTO DELLA PROVA

- Definire ed implementare il tipo di dato astratto richiesto e le relative funzioni nei file `compito.h` e `compito.cpp`. Il file `main.cpp` contiene la funzione principale `main()` ed è utilizzato dallo studente per testare la sua implementazione della classe. Il file `main.cpp` può essere modificato a piacere. In sede di valutazione dell’elaborato verrà considerato **esclusivamente il contenuto dei file `compito.h` e `compito.cpp`** ed è pertanto **vietato cambiare nome a tali file**.

Per compilare e linkare dare il comando:

```
$ g++ main.cpp compito.cpp (eseguibile invocabile tramite $ ./a.out)
```

(utilizzare `g++ -g` per includere le informazioni di debug qualora si intenda debuggare con `ddd`).

PER CONSEGNARE O RITIRARSI

Recarsi dal docente avendo preso nota dell’identificativo della macchina (g34, s23, ...).

REGOLA PER LA CORREZIONE

Per il superamento della prova pratica è **obbligatorio** aver definito ed implementato correttamente la classe e le seguenti funzioni:

- costruttore (`BattagliaNavale b(3,7,"terza partita")`)
- posiziona una nave (`b.posiziona('A',4)`)
- colpisce una posizione (`b.colpisci('C',2)`)
- stampa a video del cinema (`cout<<b`)

Il codice di altre funzioni che impedisca la compilazione deve essere messo come commento. Tali funzioni verranno comunque valutate in sede di correzione della prova.