

FONDAMENTI DI INFORMATICA I

FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI

Una avvitatore robotico utilizza per il suo funzionamento un `CaricatoreDiViti`, ossia un contenitore fisico di viti dotato di una capienza massima prefissata. Il caricatore prima di poter essere utilizzato viene caricato *dal basso* in maniera parziale o completa con le viti necessarie all'assemblaggio dello specifico componente presente sulla catena di montaggio. Le viti sono solo di due tipi: *lunghe* (L) e *corte* (C). Una volta che il caricatore è stato caricato, esso può essere utilizzato dal robot che avvita tutte le viti. Le operazioni che si possono effettuare sul caricatore di viti sono le seguenti:



(PRIMA PARTE)

- ✓ **`CaricatoreDiViti c;`**
Costruttore di default, che inizializza una caricatore `c` avente capienza pari a 10 viti. Inizialmente il caricatore non contiene viti.
- ✓ **`CaricatoreDiViti c(N);`**
Costruttore di conversione, che consente di usare un *int* ovunque occorre un caricatore. Crea un caricatore vuoto con capienza pari a N viti.
- ✓ **`c+=p;`**
Operazione che aggiunge al più `p` viti *lunghe* al caricatore, fino alla capienza massima. Le eventuali viti in eccesso non vengono inserite.
- ✓ **`c*=p;`**
Operazione che aggiunge al più `p` viti *corte* al caricatore, fino alla capienza massima. Le eventuali viti in eccesso non vengono inserite.
- ✓ **`cout << c`**
Operatore di uscita per il tipo `CaricatoreDiViti`. L'uscita ha la forma seguente:

```
-
-
L
L
L
C
C
C
C
C
```

In questo esempio il caricatore `c` è di dimensione 10 e sono state inserite prima 3 viti lunghe e successivamente 5 corte.

(SECONDA PARTE)

- ✓ **`c.avvitaTutte()`**
Operazione che avvita tutte le viti presenti nel caricatore, provocandone lo svuotamento.
- ✓ **`CaricatoreDiViti c1(c)`**
Costruttore di copia che inizializza un oggetto `c1` con il valore dell'oggetto `c`.
- ✓ **`operator int(c)`**
Operatore di conversione, che consente di usare un contenitore ovunque occorre un *int*. L'operatore ritorna il numero di viti presenti nel caricatore.
- ✓ **`~CaricatoreDiViti()`**
Distruttore della classe `CaricatoreDiViti`.

Mediante il Linguaggio C++, realizzare il tipo di dati astratti , definito dalle precedenti specifiche, rappresentando un `CaricatoreDiViti` *mediante unsigned interpretati come configurazione di bit (vettori di bit)*. Individuare le 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 abbiamo 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 di default (`CaricatoreDiViti c;`)
- operatore `+=` che aggiunge `p` viti lunghe (`c+=p;`)
- operatore `*=` che aggiunge `p` viti corte (`c*=p;`)
- stampa a video (`cout<<c`)

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.