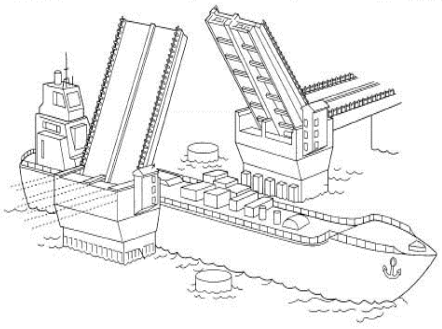
FONDAMENTI DI INFORMATICA I **PROVA PRATICA 17 GENNAIO 2015**
FOND. DI INFORMATICA E PROGRAMMAZIONE A OGGETTI
CORSO DI LAUREA IN INGEGNERIA INFORMATICA



Un Ponte Mobile è un particolare tipo di ponte che, quando è chiuso, consente alle automobili di attraversare il fiume come un qualunque altro normale ponte. A differenza dei comuni ponti esso è apribile: quando è aperto, esso consente il transito alle navi (in questo stato il transito delle automobili è precluso). In particolare, quando il Ponte Mobile è aperto le automobili si pongono in attesa in base all'ordine di arrivo. Quando il ponte si chiude, tutte le auto in attesa vengono fatte passare. Quando il ponte è chiuso non ci sono auto in attesa. Ogni auto è identificata in modo univoco da una stringa di al più 6 lettere maiuscole dell'alfabeto.

Implementare le seguenti operazioni che possono essere compiute su un `PonteMobile`:

PRIMA PARTE ***(qualora siano presenti errori di compilazione, collegamento o esecuzione in questa parte, l'intera prova verrà considerata insufficiente e pertanto non verrà corretta)***

✓ `PonteMobile p(s);`
Costruttore che crea un ponte mobile. Il ponte è aperto ed è presente un'automobile con identificatore `s` in attesa di fronte a `p`.

✓ `p.aggiungi(s);`
Operazione che aggiunge l'auto con identificatore `s` a quelle in attesa al ponte `p`. Se il ponte è chiuso, l'auto viene fatta passare. Se il ponte è aperto l'auto viene messa in attesa. NB: se un'auto con lo stesso identificatore esiste già in attesa, **non** viene effettuata l'aggiunta della nuova auto. In quest'ultimo caso il ponte rimane inalterato (*non sono ammessi identificatori duplicati*).

✓ `cout<<p;`
Operatore di uscita del ponte mobile. Innanzitutto viene mostrato lo stato del ponte:
`PONTE APERTO` o `PONTE CHIUSO`.
Nel caso di ponte aperto, vengono anche mostrati gli identificatori delle auto in attesa **in ordine di arrivo** (premettere i caratteri "=>" ad ogni identificatore). Nell'esempio seguente il ponte p è aperto e ci sono 3 auto in attesa. L'identificatore dell'auto arrivata per prima è AAA, quello dell'auto arrivata per ultima è C.

```
PONTE APERTO=>AAA=>BB=>C
```

SECONDA PARTE ***(si invita a mettere sotto commento le operazioni di questa seconda parte che dovessero impedire la compilazione, il collegamento o la corretta esecuzione del codice)***

✓ `p.cambiaStato();`
Operazione che modifica lo stato del ponte. Se il ponte cambia stato da aperto a chiuso, l'operazione fa passare tutte le auto in attesa.

✓ `p -= s;`
Operatore di sottrazione e assegnamento che modifica il ponte mobile `p` eliminando l'auto con identificatore `s` da quelle in attesa. Se l'auto non è presente, il ponte mobile rimane inalterato.

✓ `~p;`
Operatore di complemento bit a bit che restituisce il numero di auto in attesa di fronte al ponte `p`.

✓ `~PonteMobile();`
Distruttore.

Mediante il linguaggio C++, realizzare il tipo di dato astratto **`PonteMobile`**, 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 al prompt spostarsi sulla cartella 'elaborato' (`$ cd ~/elaborato`). Si utilizzi il comando `pwd` per verificare che ci si trovi nella cartella corretta `/home/studenti/elaborato`.

• Sempre al prompt 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, dopodiché 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 **dopo aver preso nota dell'identificativo della macchina** (esempi: g34, s23, c22, …).

USCITA CHE DEVE PRODURRE IL PROGRAMMA

```
Test costruttore e op. di uscita (deve stampare 'PONTE APERTO=>AAA')
PONTE APERTO=>AAA

Test della aggiungi (deve stampare 'PONTE APERTO=>AAA=>BB=>C')
PONTE APERTO=>AAA=>BB=>C

Altro test della aggiungi e dell'op. di compl. bit a bit (deve stampare 3)
3

Test operatore -= (deve stampare 'PONTE APERTO=>AAA=>C')
PONTE APERTO=>AAA=>C

Test della cambiaStato (deve stampare 'PONTE CHIUSO')
PONTE CHIUSO
```