FONDAMENTI DI INFORMATICA I **PROVA PRATICA 06 LUGLIO 2015**

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Un `Ristorante` ha `N` posti a sedere e serve gruppi di clienti. I gruppi di clienti che arrivano al ristorante sono fatti accomodare fintantoché ci sono posti disponibili; altrimenti vengono fatti aspettare in coda. Per ogni gruppo inserito in coda, viene memorizzato il cognome di uno dei componenti ed il numero di persone che lo compongono. Si assuma che tutti i cognomi siano diversi e che siano composti da 30 caratteri al massimo. Implementare il tipo di dato astratto `Ristorante`, dotandolo delle seguenti operazioni:

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

- ✓ **`Ristorante r(N);`**

Costruttore che crea un ristorante con `N` posti a sedere. Inizialmente non ci sono gruppi di clienti in attesa e tutti i posti a sedere sono liberi.

- ✓ **`r.aggiungi(cg,num);`**

Operazione che registra l'arrivo di un gruppo di clienti. Il gruppo è identificato dal cognome `cg` ed è composto da `num` persone (`num` si assuma <= `N`). Se `num` è minore o uguale al numero di posti liberi, il gruppo `cg` viene fatto accomodare (***e l'informazione relativa al cognome viene scartata***). Altrimenti viene inserito in coda agli eventuali gruppi arrivati precedentemente ad esso e non ancora serviti. ***NB: i componenti di uno stesso gruppo non debbono necessariamente essere fatti accomodare in posti vicini.***

- ✓ **`cout << r;`**

Operatore di uscita per il tipo `Ristorante`. I gruppi in coda vengono stampati nell'ordine in cui sono arrivati. Nel caso di un ristorante di 14 posti con 2 posti liberi e i seguenti 4 gruppi in attesa:
`Verdi (4 persone), Bianchi (6 persone), Neri (2 persone) e Rossini (5 persone)`
l'uscita dovrà essere la seguente (nell'ipotesi che il gruppo Rossini sia arrivato per ultimo):

```
Posti liberi 2, in attesa Verdi(4) Bianchi(6) Neri(2) Rossini(5)
```

Nel caso in cui il medesimo ristorante fosse vuoto, l'uscita sarebbe:

```
Posti liberi 14, in attesa nessuno
```

- ✓ **`r-=(num);`**

Operazione che libera `num` posti a sedere. Se ci sono gruppi in attesa, questi vengono fatti accomodare a sedere secondo l'ordine di arrivo e finché ci sono posti disponibili.

--- SECONDA PARTE ---

- ✓ **`Ristorante r2(r);`**

Costruttore di copia che inizializza `r2` con il valore di `r`.

- ✓ **`r.rinuncia(cg);`**

Operazione che cancella dalla coda il gruppo identificato dal cognome `cg`. La funzione restituisce `false` se non è presente nella coda un gruppo identificato da `cg`; altrimenti restituisce `true`. NB: qualora a seguito della rinuncia di un gruppo uno o più gruppi in attesa possano essere fatti accomodare, la rinuncia provvede a farlo. **Esempio**: Qualora lo stato attuale del ristorante sia:

```
Posti liberi 5, in attesa Bianchi(6) Neri(2) Rossini(5)
```

l'eventuale rinuncia di `Bianchi` deve far si che il gruppo Neri venga fatto accomodare:

```
Posti liberi 3, in attesa Rossini(5)
```

- ✓ **`~Ristorante();`**

Distruttore.

Mediante il Linguaggio C++, realizzare il tipo di dato astratto **`Ristorante`**, definito dalle precedenti specifiche. **Gestire le eventuali situazioni di errore**.

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 abbiano 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. uscita. Deve stampare: 'Posti liberi 14, in attesa nessuno'
Posti liberi 14, in attesa nessuno

Test della 'aggiungi'. Deve stampare: 'Posti liberi 2, in attesa  Verdi(4) Bianchi(6) Neri(2) Rossini(5)'
Posti liberi 2, in attesa Verdi(4) Bianchi(6) Neri(2) Rossini(5)

Testa dell'op. -= (7). Deve stampare: 'Posti liberi 5, in attesa Bianchi(6) Neri(2) Verdini(5)'
Posti liberi 5, in attesa Bianchi(6) Neri(2) Rossini(5)

Test del costruttore di copia. Deve stampare 'Posti liberi 5, in attesa Bianchi(6) Neri(2) Rossini(5)'
Posti liberi 5, in attesa Bianchi(6) Neri(2) Rossini(5)

Test del distruttore (r2 sta per essere distrutto): non deve stampare nulla

Test della 'rinuncia'. Deve stampare: 'Bianchi ha rinunciato. Verdini non trovato'
Bianchi ha rinunciato. Verdini non trovato

Altra verifica della 'rinuncia'. Deve stampare: 'Posti liberi 3, in attesa Rossini(5)'
Posti liberi 3, in attesa Rossini(5)
```