

Una pizzeria serve gruppi di clienti ed ha a disposizione **N** posti a sedere. I gruppi di clienti che arrivano in pizzeria vengono fatti accomodare fintantoché risultano 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. Si supponga di realizzare una classe **GestPiz** per la gestione dei clienti.

Le operazioni che possono essere effettuate sul tipo **GestPiz** sono le seguenti:

- **GestPiz g(N)**
Costruttore che gestisce una pizzeria con **N** posti a sedere. Inizialmente non ci sono gruppi di clienti in attesa e tutti i posti a sedere sono liberi.
- **GestPiz g1(g)**
Costruttore di copia che inizializza g1 con il valore di g.
- **~GestPiz()**
Distruttore.
- **g.arriva(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. Se non ci sono già altri gruppi in attesa e nella pizzeria ci sono almeno **num** posti liberi, il gruppo viene fatto accomodare in sala; altrimenti viene inserito in coda.
- **g.libera(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.
- **g.cancella(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**.
- **cout << g**
L'operatore di uscita per il tipo **GestPiz**. I gruppi in coda vengono stampati nell'ordine in cui sono arrivati. Il formato dell'uscita è il seguente:

```
Posti Liberi: 2
Gatti 5
Lupi 1
Ceri 2
```

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti **GestPiz**, definito dalle precedenti specifiche.

NOTE SULLO SVOLGIMENTO DELLA PROVA PRATICA:

- Effettuare il **login**
Nome: studenti
Password: studenti
- Aprire il *Dev-C++* (dal Menù *Avvio* (o *Start*) nella barra degli strumenti in fondo allo schermo, selezionare *Programmi* e quindi *Dev-C++*)
- **Prima di iniziare a svolgere l'elaborato**, selezionare la voce *Identifica studente* nel menù *Strumenti* all'interno dell'ambiente *Dev-C++* e inserire i dati richiesti
- Dal menu *File* del *Dev-C++*, aprire il progetto *esainf.dev* presente nella cartella *c:\esame\esaInf*. Il progetto contiene tre file, denominati *compito.h*, *compito.cpp* e *main.cpp*
- Scrivere la dichiarazione della classe nel file *compito.h* e la definizione delle funzioni nel file *compito.cpp*. Il file *main.cpp* contiene la funzione principale *main()* che serve a verificare le funzioni scritte nel file *compito.cpp*. Il file *main.cpp* può essere modificato. **Si tenga presente, comunque, che in sede di valutazione dell'elaborato verrà considerato esclusivamente il contenuto dei file *compito.h* e *compito.cpp***
- **Per una corretta stampa dell'elaborato**, non scrivere righe di codice di lunghezza eccessiva (mantenersi entro i margini imposti dall'ambiente *Dev-C++* cioè entro la linea verticale presente alla destra della pagina);
- **Per la Consegna:**
 - Selezionare la voce *Consegna* dal menù *Strumenti* (o *Tools*) all'interno dell'ambiente *Dev-C++* e premere il tasto INVIO fino a quando non viene chiusa la finestra che è stata attivata.
 - Aspettare al proprio posto di essere chiamati per verificare la stampa del proprio elaborato, firmarlo e **consegnarlo definitivamente**.

Condizione necessaria per la correzione dell'elaborato è che le prime tre funzioni siano state implementate correttamente.