

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 $\leq N$). 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 sì che il gruppo `Neri` venga fatto accomodare:

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

✓ **`~Ristorante()`** ;

Distruttore.