

Smistacasse è un sistema che gestisce l'allocazione alle casse dei clienti di un supermercato. Ogni cliente viene identificato univocamente da un numero intero diverso da zero e dal numero di articoli acquistati. Le casse sono numerate a partire da 1. Una cassa può essere aperta o chiusa. Il numero di articoli in coda ad una cassa è pari alla somma degli articoli dei clienti in coda a quella cassa.

L'allocazione dei clienti alle casse viene scelta in base alla cassa con il minimo numero di articoli in coda in modo da minimizzare i tempi di attesa dei clienti.

Implementare le seguenti operazioni che possono essere fatte su Smistacasse:

--- **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)*

✓ **Smistacasse s(N) ;**

Costruttore che crea un sistema Smistacasse con N casse (≥ 1). Tutte le casse sono aperte. Le code di tutte le casse sono vuote.

✓ **s.trovaCassa() ;**

Operazione che ritorna il numero della prima cassa col minor numero di articoli in coda.

✓ **s.aggiungi(id, n) ;**

Operazione che aggiunge il cliente avente identificatore `id`, con numero di articoli acquistati `n`, al sistema `s`. Non possono esistere clienti nel sistema con lo stesso identificatore.

✓ **cout << s ;**

Operatore di uscita per il tipo Smistacasse. L'uscita per un sistema Smistacasse con 3 casse, dopo l'arrivo del cliente con `ID=68` con 11 articoli e del cliente con `ID=74` con 7 articoli ha il seguente formato:

1: `(ID=68, ARTICOLI=11)` .

2: `(ID=74, ARTICOLI=7)` .

3: `<vuota>`

L'uscita, dopo l'arrivo di altri due clienti (`ID=76` con 8 articoli e `ID=52` con 9 articoli, rispettivamente) ha il seguente formato:

1: `(ID=68, ARTICOLI=11)` .

2: `(ID=74, ARTICOLI=7) (ID=52, ARTICOLI=9)` .

3: `(ID=76, ARTICOLI=8)` .

All'arrivo del terzo cliente la terza cassa viene impegnata, infine il quarto cliente viene messo in coda alla seconda cassa perchè quella con meno articoli da battere.

✓ **s.servi(nCassa) ;**

Operazione che serve il primo cliente in coda alla cassa `nCassa`. Lascia lo Smistacasse invariato qualora non ci siano clienti in coda a quella cassa.

--- SECONDA PARTE ---

✓ `s-=(nCassa);`

Operazione che chiude la cassa `nCassa`. I clienti in coda presso quella cassa devono essere smistati tra le casse rimanenti **con la stessa politica utilizzata per l'arrivo di un nuovo cliente**.

La stampa a video di una cassa chiusa ha il seguente formato:

```
1: <chiusa>
2: (ID=52, ARTICOLI=9) .
3: (ID=68, ARTICOLI=11) .
```

✓ `~Smistacasse();`

Distruttore.

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

USCITA CHE DEVE PRODURRE IL PROGRAMMA

--- PRIMA PARTE ---

Test del costruttore e dell'operatore di uscita

```
1: <vuota>
2: <vuota>
3: <vuota>
```

Test della 'aggiungi'

```
1: (ID=68, ARTICOLI=11) .
2: (ID=74, ARTICOLI=7) .
3: <vuota>
```

```
1: (ID=68, ARTICOLI=11) .
2: (ID=74, ARTICOLI=7) (ID=52, ARTICOLI=9) .
3: (ID=76, ARTICOLI=8) .
```

Test della 'trovaCassa'

```
3
```

Test della 'servi' sulla seconda e terza cassa

```
1: (ID=68, ARTICOLI=11) .
2: (ID=52, ARTICOLI=9) .
3: <vuota>
```

--- SECONDA PARTE ---

Test dell'operatore `-=` sulla prima cassa

```
1: <chiusa>
2: (ID=52, ARTICOLI=9) .
3: (ID=68, ARTICOLI=11) .
```

Test del distruttore (`s1` e' stata appena distrutta)