

Un `Cinema` è identificato da una stringa di lettere minuscole o maiuscole dell'alfabeto di lunghezza minore o uguale a 30 e mantiene informazioni sulle prenotazioni dei posti della sala. La sala del cinema è composta da un numero di file minore o uguale a 26 (indicate da lettere maiuscole dell'alfabeto: A è la prima fila, B è la seconda fila e così via) ed ogni fila ha un numero di posti minore o uguale a 9 (numerati con numeri interi a partire da

1). Implementare le seguenti operazioni che possono essere effettuate su un `Cinema`:

`Cinema c(n, p, s)`

Costruttore che inizializza un `Cinema c` con nome indicato dalla stringa `s`, `n` file e `p` posti per fila. Inizialmente tutti i posti sono liberi.

`c.prenota(l, j)`

Funzione che prenota il posto `j` della fila `l`, dove `j` è un intero ed `l` è un carattere. Se il posto è già occupato, la funzione lascia il cinema inalterato e restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.

`c.cancella(l, j)`

Funzione che cancella la prenotazione del posto `j` della fila `l`. Se il posto è libero, la funzione lascia il cinema inalterato e restituisce `false`. Se l'operazione ha successo la funzione restituisce `true`.

`cout << c`

Operatore di uscita per il tipo `Cinema`. Viene stampato il nome del cinema e, per ogni fila, per ogni posto viene visualizzato se il posto è libero o prenotato secondo il formato seguente:

```
Nome del cinema: Moderno
```

```
 1234567
A  *
B
C*****
*
```

In questo esempio, il nome del cinema è "Moderno", la sala ha 3 file e 7 posti per fila. Nella prima fila tutti i posti sono liberi eccetto il quarto; nella seconda fila tutti i posti sono liberi; nella terza fila tutti i posti sono prenotati.

`!c`

Operatore di negazione logica che restituisce il numero di posti liberi della sala.

`c1=c2`

Operatore di assegnamento.

`bool prenotaGruppo(k)`

Funzione che prenota i primi `k` posti adiacenti liberi nella stessa fila, a partire dalla prima fila e dal primo posto. Se l'operazione fallisce, la funzione restituisce `false` e lascia il cinema inalterato. Altrimenti restituisce `true`.

`~Cinema ()`

Distruttore.

Utilizzando il linguaggio C++, realizzare il tipo di dati astratti `Cinema` definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento. Si faccia riferimento al main riportato a pagina 2.

```

// file main.cpp
#include <iostream>
#include "compito.h"
using namespace std;

int main(){
    // test prima parte
    Cinema c(3,7,"Moderno");
    c.prenota('A',4);
    c.prenota('C',1);
    c.prenota('C',2);
    c.prenota('C',3);
    c.prenota('C',4);
    c.prenota('C',5);
    c.prenota('C',6);
    c.prenota('C',7);
    c.prenota('B',4);
    cout<<c;
    c.cancella('B',4);
    cout<<c;
    // test seconda parte
    {
        Cinema c2(5,9,"Odeon");
        cout<<c2;
        c2 = c;
        c2.prenota('B',2);
        cout<<"Il numero di posti liberi in c2 e' :"<<!c2<<endl;
        if(c.prenotaGruppo(4))
            cout<<c2;
        else
            cout<<"Impossibile prenotare per il gruppo di 4 persone"<<endl;
    }
    return 0;
}

```

Uscita attesa

Nome del cinema: Moderno

```

1234567
A *
B *
C*****

```

Nome del cinema: Moderno

```

1234567
A *
B
C*****

```

Nome del cinema: Odeon

```

123456789
A
B
C
D
E
Il numero di posti liberi in c2 e' :12

```

Nome del cinema: Moderno

```

1234567
A *
B *
C*****

```