

- Você foi contratado para auxiliar em um sistema de um cinema que controlará venda de ingressos para ocupação das poltronas de uma sessão.

A identificação das poltronas segue a sequência estabelecida pelas filas existentes e suas respectivas colunas (corredores).

Hoje a sala do cinema possui 16 filas, que tem a numeração dividida de 0 a 15 e (a fila 0 representa o corredor superior).

Para cada fila, temos 20 poltronas, sendo que 10 poltronas (indicadas de 0 a 9) estão localizadas à esquerda e 10 poltronas (indicadas de 11 a 20) localizadas à direita.

A coluna 10 representa o corredor central.

Para entendimento gráfico, vamos imaginar o mapa da sala de cinema para a alocação das reservas como se fosse uma matriz bidimensional (utiliza um tipo abstrato de dados para representar este tipo de aplicação).

-	0	...	9	10	11	...	20
0							
1	LADO ESQUERDO			CORREDOR	LADO DIREITO		
...							
15							

Verificando uma alocação, dizer que [8,11] (Considerar [linha,coluna]) identifica a poltrona número 11 do lado esquerdo da fila 8.

Para identificar que a poltrona[8,11] está ocupada consideramos o valor armazenado como sendo 1, ou seja, poltrona[8,11]=1; caso contrário ela receberá 0, ou seja, poltrona[8,11] = 0 identificando que está livre.

Para qualquer posição a qual a coluna é 10 (poltrona[i,10]) e, qualquer posição a qual a fila é 0 (poltrona[0,j]), consideramos com caso especial por se tratar do corredor, sendo assim não poderá receber valor 0, sempre será 1.

Poderíamos entender a matriz RESERVA como da seguinte forma e considerar utilizar este tipo dentro de uma estrutura:

```
typedef int CINEMA[16][21];
```

```
CINEMA RESERVA;
```

Atualmente, para se reservar uma poltrona, inicialmente faz-se uma verificação para ver se a mesma está desocupada e em caso positivo esta é reservada, caso contrário é solicitado ao interessado que informe outra poltrona e o procedimento é repetido.

Por exemplo:

- Reservar a poltrona i da coluna j , faz-se uma consulta.
- caso $RESERVA[i, j] = 1$, deve-se escolher outro lugar;
- caso $RESERVA[i, j] = 0$, a reserva é concedida e o mapa alterado através da atribuição $RESERVA[i, j] = 1$.

A partir então com o cenário descrito faça:

- Implemente as estruturas necessárias para a criação do sistema para esta situação.
 - Crie um procedimento ou função para realizar uma reserva no cinema (considerar a poltrona não estar reservada).
 - Crie um procedimento ou função para cancelar uma reserva no cinema (considerar que a poltrona está reservada).
 - Crie um procedimento ou função para verificar se uma poltrona está reservada ou não.
 - Criar um procedimento ou função para mostrar todas as poltronas indicando se estão reservadas ou não.
 - Crie um procedimento ou função para mostrar todas as poltronas que estão reservadas.
 - Crie um procedimento ou função para mostrar todas as poltronas que estão livres.
2. Suponha agora que irão ocorrer duas sessões de cinema a cada dia. Assim precisaremos de uma estrutura para a matriz $RESERVA$ tridimensional, a qual consideraremos um novo índice k que será 1 ou 2, conforme a sessão desejada.

Poderíamos considerar a implementação desta matriz como sendo

```
typedef int CINEMASESSAO[2][16][21];
```

```
CINEMASESSAO RESERVAESSAO;
```

A partir então com o cenário atual descrito faça:

- Realiza o mesmo apresentado nas letras de a) a g) da questão 1 considerando que teremos 2 sessões no mesmo dia.