

# Trabalho Prático de PC2

**Grupo: David Cubric DRE: 122148485**

**Victor F. Ferreira DRE: 122138901**

**Carlos Eduardo DRE: 122121661**

**Professora: Valéria Bastos**

## Constantes

#N\_FILMES 3

#VALOR\_INGRESSO 20

#COLUNA 10

#FILEIRA 40

## Structs

Enumeração que representa os filmes disponíveis na cinema

```
typedef enum _Filmes {
```

```
    VELOZES_E_FURIOSOS_137 = 1,
```

```
    THE_FLASH,
```

```
    TRANSFORMERS
```

```
} Filmes;
```

## Bibliotecas

<stdio.h>

<stdlib.h>

<stdbool.h>

<ctype.h>

## Funções

### Requisitos 1 e 13

#### *1 - Filmes seleciona\_filme()*

Essa função perguntará ao usuário para qual filme deseja assistir, cada um dos 3 filmes é representado pelo número da sala em que o mesmo será exibido(1 - Star trek, 2 - Star wars, 3 - Avengers, 0 - Encerrar). O usuário deverá digitar um número inteiro de 0 a 3, caso o valor digitado seja maior que 3 ou menor que 0 o programa exibirá na tela que o valor digitado é inválido e que o usuário deve digitar outro valor que seja válido. Após digitar um valor válido a função retornará esse valor à main. Caso o valor retornado seja 0 o programa encerrará.

### Requisito 2

#### *2 - int compra\_de\_ingressos(int sala\_escolhida)*

A função perguntará ao usuário quantos ingressos deseja comprar para o filme que foi escolhido (indicado pelo parâmetro sala). O usuário deverá digitar um número inteiro de 1 a 400, caso o valor seja maior que 400 será exibido que a quantidade desejada não está disponível, caso o valor seja menor que 1 será exibido que essa quantidade não é válida. Enquanto a quantidade de ingressos for um valor indisponível ou um valor inválido a função pedirá uma nova quantidade de ingressos. A função retornará a quantidade total de ingressos que serão comprados.

#### *3- int confirma\_fime()*

Após ser apresentado os locais disponíveis para a sessão, o usuário

deve digitar 1 para continuar ou 0 para cancelar a operação.

### **Requisito 3**

*4 - int desconto\_meia\_entrada(int ingressos\_totais)*

A função recebe como parâmetro a quantidade de ingressos que o usuário deseja adquirir e perguntará ao usuário quantos ingressos vão receber o desconto meia entrada. Enquanto a quantidade de ingressos meia entrada for superior a quantidade de ingressos totais, o usuário deverá digitar novamente a quantidade de ingressos que receberão o desconto meia entrada. A função retornará a quantidade de ingressos que terão o desconto meia entrada.

### **Requisitos 4 e 8**

*5- bool valida\_carteira(int carteira)*

A função recebe um número de carteira como argumento e verifica se ele é válido. O algoritmo de validação é baseado na soma dos dígitos do número de carteira. A função retorna true se o número de carteira é válido e falso caso contrário.

*6- int valida\_meia\_entrada(int qtd\_meia\_entrada)*

A função recebe a quantidade de ingressos que receberão o desconto de meia-entrada, solicita ao usuário que digite o número de carteira de estudante para cada ingresso, valida as carteiras de estudante inseridas e exibe mensagens adequadas ao resultado da validação. A função mantém um registro das carteiras de estudante utilizadas para evitar duplicação. Se o usuário optar por cancelar o desconto de um ingresso, a função exibe uma mensagem correspondente. A função retorna 0 se a validação for concluída com sucesso ou -1 se não houver ingressos de meia-entrada. A função utiliza a função valida\_carteira() para validar os números de carteira de estudante.

## Requisitos 5

*7 - Int qtd\_itasil(int qtd\_meia\_entrada, int qtd\_ingressos\_totais)*

A função recebe a quantidade de ingressos meia-entrada, a quantidade total de ingressos e pergunta ao usuário quantos ingressos recebem o desconto itasil. Enquanto a quantidade de ingressos itasil mais a quantidade de ingressos meia-entrada for maior do que a quantidade de ingressos totais, o usuário recebe uma mensagem avisando que a quantidade de ingressos que recebem o desconto itasil é inválida. O usuário deve digitar a quantidade novamente ou cancelar o desconto, após digitar um valor válido é exibido na tela a quantidade de ingressos que recebem o desconto itasil. A função retorna essa quantidade de ingressos.

## Requisitos 6 e 7

*8 - int busca\_codigo\_itasil( int clientes\_itasil[ ], int tamanho\_vetor\_itasil, int codigo\_cliente\_itasil)*

*A função recebe o vetor clientes\_itasil, o tamanho desse vetor e o código que será buscado por meio de busca sequencial. Ela retorna 1 se o código buscado estiver contido no vetor e -1 caso ele não esteja.*

*9 - int valida\_cliente\_itasil(int qtd\_cliente\_itasil, int cliente\_itasil[ ])*

A função recebe a quantidade de ingressos que recebem o desconto itasil e recebe o vetor cliente\_itasil para validar e armazenar os códigos desses clientes. O usuário deverá entrar com um código de cliente válido, ou seja, um número positivo que seja múltiplo de 341 e de 001 ou o valor 0 para cancelar o desconto. Caso o código não seja múltiplo de 341 ou de 001 a função exibirá que o código não é válido e pedirá ao usuário por um novo código ou o valor 0 para cancelar o desconto. Além disso, por meio da função busca\_codigo\_itasil e o vetor que armazena os códigos, um mesmo código de cliente itasil só poderá ser utilizado uma vez por programa, então ao tentar utilizar um mesmo código a função exibirá que o código já foi utilizado. Em ambos casos o usuário poderá digitar o código novamente ou digitar 0 para cancelar o desconto do ingresso.

## Requisito 9

*10 - void inicializa(char sala[coluna][fileira])*

A função inicializa a matriz com o valor 0, indicando que todos os assentos estão vazios.

*11 - void exhibe\_mapa(char sala[coluna][fileira])*

A função exibirá uma matriz de caracteres que representará o mapa da sala referente ao filme escolhido de cinema.

ex: mapa inicial de uma das 3 salas.

## Requisitos 10 e 11

*12 - void reserva\_assento(int n\_ingressos\_a\_reservar, char sala [COLUNA][FILEIRA])*

A função receberá como parâmetro a matriz que representa o mapa da sala e utilizará uma estrutura de repetição para marcar a posição na sala desejada pelo usuário, de modo que seja exibido na tela que a poltrona do ingresso x/n deve ser selecionada (x é inicializado como 1 para representar o primeiro ingresso e é incrementado ao final do loop até que x seja igual ao número total de ingressos(n)). O usuário deverá entrar com um número inteiro de 1 a 40 (fileira) pular um espaço e entrar com uma letra de A a J (coluna), caso o usuário entre com um número menor do que 1, maior 40, não pule espaço entre o valor da fileira e o da coluna ou entre com um caractere diferente dos caracteres: A,B,C,D,E,F,G,H,I,J a função exibirá que a posição é inválida e pedirá um nova posição. Enquanto o conteúdo da coordenada informada pelo usuário for diferente de 0 a função exibirá ao cliente que o assento está ocupado

Após receber e armazenar a posição o valor dessa coordenada na matriz será alterado de 0 para X, indicando que o assento está reservado.

## Requisito 12

*13 - void calcula\_exibe\_preco(int ingressos\_inteira, int meia\_estudante, int meia\_itasil)*

A função calcula o preço que cada usuário deve pagar pela compra de todos os ingressos. O preço a pagar pelos ingressos meia estudante:  $(\text{VALOR\_INGRESSO} * \text{meia\_estudante} * 0.5)$ . O Preço a pagar pelos ingressos com desconto itasil:  $(\text{VALOR\_INGRESSO} * \text{cliente\_itasil} * 0.7)$ . Preço a pagar pelos ingressos inteira:  $(\text{VALOR\_INGRESSO} * \text{ingressos\_inteira})$ . O preço total é dado pela soma dos valores dos ingressos inteira, meia estudante e ingressos com desconto itasil.

Após o cálculo, o valor total é exibido para o usuário e ao término da transação a função exibirá uma mensagem de agradecimento pela compra.

*14 - char \*nome\_filme(Filmes filme)*

A função converte um valor da enumeração Filmes para uma string representando o nome do filme ou NULL caso o valor passado não esteja na enumeração

*15 - int confirma\_filme();*

Pede ao usuário para confirmar a escolha do filme. Caso o usuário digite 1, o filme é confirmado e o programa continua. Caso o usuário digite 0, ele será levado novamente para a tela de escolha de filmes.

*16 - assento\_fora\_dos\_limites(int fileira, int coluna)*

Checa se o assento desejado na posição fileira coluna está fora dos limites do cinema. Caso o assento esteja fora dos limites, a função retornará 1, caso contrário retornará 0

*17 - int assento\_ocupado(int fileira, int coluna, char sala[COLUNA][FILEIRA])*

Checa se o assento na posição sala[coluna][fileira] está ocupado. A função retorna 1 caso o assento esteja ocupado e 0 caso contrário

*18 - void carrega\_sala(Filmes filme, char sala[COLUNA][FILEIRA])*

Lê um arquivo contendo os estados dos assentos de uma sala e o

armazena na variável sala. O arquivo a ser lido é definido pelo valor da enumeração passada

*19 - void salvar\_sala(Filmes filme, char sala[COLUNA][FILEIRA])*

Escreve o estado de uma sala dentro do arquivo que representa os estados dos assentos do filme que representa

## **Estruturas de Dados Utilizadas**

- O programa utiliza uma matriz bidimensional para representar o mapa da sala
- O programa utiliza a Struct Filmes que contém as opções de filmes para o usuário.
- O programa utiliza vetor para armazenar as carteiras de estudante e garantir que cada carteira será utilizada uma única vez por programa
- O programa utiliza vetor para armazenar os códigos Itasil e garantir que cada código será utilizado uma única vez por programa

## **Utilização de arquivos**

O programa utiliza arquivos para armazenar o estado do mapa de assentos do cinema. Isso permite que entre as execuções do programa, as informações fiquem salvas.

A função `carrega_sala()` é responsável por ler o arquivo correspondente ao filme escolhido e preencher a matriz `sala` com os dados armazenados no arquivo. Essa função é chamada no início do programa para carregar o estado anterior do mapa de assentos.

A função `salvar_sala()` é chamada ao final do programa para salvar o estado atual do mapa de assentos no arquivo correspondente ao filme escolhido.

### **Chamadas às funções na main**

1-`filme_escolhido = seleciona_filme();`

2-`printf("\nIngressos disponiveis para o filme %s:\n",  
nome_filme(filme_escolhido));`

3-`carrega_sala(filme_escolhida, sala);`

4-`exibe_mapa(sala);`

5-`filme_confirmado = confirma_filme();`

6-`qtd_ingressos_totais = compra_de_ingressos(filme_escolhido,  
qtd_ingressos_restantes[filme_escolhido]);`

7-`qtd_meia_entrada = valida_meia_entrada(desconto_meia_entrada  
(qtd_ingressos_totais), clientes_meia_entrada);`

8-`itasil_desconto = valida_cliente_itasil(qtd_itasil(qtd_meia_entrada,  
qtd_ingressos_totais), cliente_itasil);`

9-`carrega_sala(filme_escolhido, sala);`

10-`exibe_mapa(sala);`

11-`reserva_assento(qtd_ingressos_totais, sala);`



12-exibe\_mapa(sala);

13-salvar\_sala(filme\_escolhido, sala);

14-calcula\_exibe\_preco((qtd\_ingressos\_totais - qtd\_meia\_entrada -  
itasil\_desconto), qtd\_meia\_entrada, itasil\_desconto);