

Este exame vale 100 pontos + questão bônus de 10 pontos.

1. Exercício de programação - No Multiverso das Estruturas... (110 pts)

Nesta semana estreia o mais novo filme da Marvel, “Doutor Estranho no Multiverso da Loucura”, e você, sendo um ávido fã da nona arte desde a infância, pretende assistir o filme na estreia para evitar *spoilers*. Como você estará resolvendo esta prova na quarta-feira, dia da pré-estreia, acabou decidindo ir na quinta-feira à noite. Para tal, você fará um levantamento de algumas sessões de cinema previstas para quinta-feira aqui na capital antes de decidir qual sessão comparecer.



Cinema	Sessão				Sessão			
	Hora	Idioma	3D	Preço	Hora	Idioma	3D	Preço
Cinemark Barra Sul	20	L	0	38,00	21	L	1	46,00
GNC Cinemas Iguatemi	20	D	0	26,00	22	L	0	26,00
Espaço de Cinema Bourbon Country	20	L	0	33,00	21	D	0	33,00
Cinemark Bourbon Ipiranga	19	D	1	38,00	21	L	0	32,00
GNC Cinemas Praia de Belas	19	D	0	26,00	22	L	1	32,00

Faça:

(a) Crie um vetor para armazenar as informações de 5 cinemas e suas sessões

- [5] i. Declare globalmente uma estrutura chamada **Sessão** para armazenar as informações de uma sessão: **hora** (int), **idioma** (char, que pode ser ‘L’ de legendado, ou ‘D’ de dublado), **3D** (pode ser 0 ou 1, indicando *não* ou *sim*, isto é, se a sessão é ou não é em 3D) e o **preço** do ingresso (float).
- [5] ii. Declare globalmente uma estrutura para armazenar as informações de um **Cinema**, contendo o **nome** do local (string) e um **vetor de duas sessões** (variável do tipo **Sessão**).
- [5] iii. Na main, declare um vetor de cinemas (isto é, vetor da estrutura criada) contendo 5 posições.

(b) Inicialize os 5 cinemas, chamando para cada um deles uma **função** que leia os dados de um cinema

- [5] i. Crie uma função “**le_cinema**”, que **não deve receber parâmetros** e **deve retornar** um cinema.
- [5] ii. Crie uma função auxiliar “**le_sessão**”, que **não deve receber parâmetros** e **deve retornar** uma estrutura do tipo **sessão**.
- [10] iii. Dentro da função “**le_sessão**”, peça para o usuário informar todas as informações da sessão. Coloque os valores em uma estrutura do tipo **sessão**, que será retornada ao fim da função.
- [10] iv. Dentro da função “**le_cinema**”, peça para o usuário primeiramente informar o nome do cinema. Depois, inicialize o vetor de sessões desse cinema chamando a função auxiliar “**le_sessão**” para cada elemento do vetor.
- [5] v. De volta na main, copie o retorno da função “**le_cinema**” no elemento correspondente do vetor de cinemas.

(c) Faça uma **função** para imprimir as informações de todas as sessões.

- [5] i. Crie uma função “**imprime_sessões**”, que deve receber **um vetor de sessões** e o **seu tamanho** como parâmetro de entrada e não deve retornar nada.
- [10] ii. Dentro da função, imprima todos os campos de cada sessão.
- [5] iii. Na main, faça um laço que imprime o nome de cada cinema e chama a função para imprimir as informações de sessões do respectivo cinema.

(d) Faça verificações para decidir qual as melhores opções dentre as disponíveis considerando todos os cinemas.

- [5] i. Crie uma função “**verifica_cinemas**”, que deve receber o **vetor de cinemas** e o **seu tamanho** como parâmetro de entrada e não deve retornar nada.
- [25] ii. Dentro da função, compute as seguintes informações, imprimindo na tela os resultados.
 - a sessão mais barata das 21h em diante
 - a sessão mais barata legendada
 - as médias de preços das sessões em 3D e das sessões em 2D
- [10] iii. (QUESTÃO BÔNUS) Além de imprimir na tela, salve as informações em um arquivo de texto chamado “sessoes.txt”.