

Instruções gerais: A prova tem o objetivo de avaliar os conceitos ensinados e trabalhados até o momento. Ela deve ser feita individualmente e sem ajuda de terceiros. Casos de cola, terceirização, ajuda para fazer a avaliação ou qualquer outro que beneficie o aluno de forma indevida serão tratados com a penalidade máxima disponível na instituição (i.e., atribuição da nota zero e abertura de processo administrativo podendo levar a expulsão). A penalidade será aplicada com igual teor a todos os envolvidos e sem distinção (inclusive a quem fornecer). Siga as regras e faça sua própria prova usando os conceitos ensinados em sala. Boa prova!

Valor: 5 pontos

Correção: a questão está dividida em 5 partes de teste, *A*, *B*, *C*, *D* e *E*, sendo que a complexidade dos casos aumenta com as letras. Os casos das letras posteriores contemplam os casos das anteriores e mais alguns. Por exemplo, os casos de *B* contemplam os de *A*, os de *C* contemplam os de *A* e *B* e, assim por diante. Portanto, se resolver a parte *E*, resolveu toda a questão. Todas as partes possuem casos de teste ocultos não fornecidos com a questão. O Anexo I descreve os tipos de casos com mais detalhes. O aluno ganha os pontos totais se obtiver o balão na parte *E* (que inclui todos os casos) e ganha 20%, 40%, 60% e 80% se passar somente até a parte *A*, *B*, *C* ou *D*, respectivamente. A questão a ser corrigida será a última enviada, dando prioridade a que estiver correta. Portanto, sempre envie a sua solução para as duas partes. Não implementar os tipos e funções (subprogramas) pedidos abaixo levará a descontos de até 50% do valor da questão.

(BOCA: P2_2021_Q2) Você viajou no tempo por engano e acabou caindo em 2002. Para sobreviver você terá que trabalhar como programador de uma videolocadora, criando um sistema de locação de filmes na linguagem que o dono da loja exige, a linguagem C. O objetivo é controlar as operações de cadastro de filmes, aluguel de fitas, devolução de fitas e consulta ao estoque. A operação de cadastro irá cadastrar um novo filme com suas fitas no estoque, a operação de aluguel irá retirar do estoque fitas de filmes específicos, e a de devolução retornará fitas para o estoque. A consulta irá mostrar os filmes disponíveis em estoque em ordem alfabética (filmes não alugados).

Atenção: Cada filme pode conter várias fitas. Ao alugar um filme não necessariamente irá esgotar o estoque daquele filme. Não é possível fazer o cadastro de filmes que já tenham o mesmo código no sistema, caso isso aconteça deverá reportar a mensagem “Filme já cadastrado no estoque”. Também não é possível alugar ou devolver um filme que não esteja cadastrado, reportando a mensagem “Filme @ nao cadastrado” em que @ corresponde ao código do filme inválido. Demais erros podem incluir: alugar um filme que não contenha fitas no estoque (“Filme ## - @@ nao disponivel no estoque. Volte mais tarde.”), em que @@ corresponde ao nome do filme, e ## ao código do filme; ou, devolver um filme que esteja com o estoque completo (“Nao e possivel devolver o filme ## - @@”), em que @@ corresponde ao nome do filme, e ## ao código do filme.

Entrada: As operações serão dadas em blocos. Cada bloco de operação se iniciará com o nome da operação em questão, e irá terminar com o caractere “#” que indica o fim da operação. A operação de cadastro se iniciará com a string “Cadastrar” e seguirá com as linhas correspondentes ao cadastro de um filme, sendo um filme por linha até o caractere de saída do bloco da operação. A formatação das linhas de cadastro é ##,@,\$\$,%%, em que ## é o código do filme (representado por um número inteiro), @@ é o nome do filme (representado por no máximo 19 caracteres), \$\$\$ é o valor do filme (representado por um número inteiro), e %% a quantidade de fitas disponíveis para aquele filme (representado por um número inteiro). Os blocos de aluguel e devolução iniciarão respectivamente com as strings “Alugar” e “Devolver” e seguirão com as linhas de entrada. Cada linha corresponderá ao código do filme em que a ação será executada. A operação de consulta ao estoque se iniciará com a string “Estoque”, e não conterá linhas adicionais, tendo o fim da operação logo após a abertura.

Saída: Cada bloco de operação contém saídas diferentes. Além dos erros explícitos anteriormente, as saídas de sucesso incluem: ao cadastrar filmes, reportar para cada filme cadastrado a mensagem “Filme cadastrado - ## - @@”, em que @@ corresponde ao nome do filme, e ## ao código do filme. Na operação de aluguel, reportar a mensagem “Total de filmes alugados ## com custo de R\$%%”, em que ## é o número de filmes efetivamente alugados (ou seja, que não apresentaram erro) e %% corresponde ao valor total do custo dos filmes alugados naquela operação. Ao devolver uma fita o programa deverá reportar a mensagem “Filme ## - @@ Devolvido!”, em que @@ corresponde ao nome do filme, e ## ao código do filme. Ao consultar o estoque, o programa deverá reportar o estoque seguindo a formatação a seguir. Uma linha iniciando com “~ESTOQUE~” seguida de todos os filmes do estoque, sendo um em cada linha. Os filmes deverão estar ordenados alfabeticamente e cada um deles será reportado com o formato: “Filme: ## - @@ Fitas em estoque: %%”, em que @@ corresponde ao nome do filme, e ## ao código do filme e, %% ao número de fitas atual do filme. A saída de cada bloco de operação será separada por uma linha em branco. Ao terminar todas as operações, o programa deverá exibir o lucro total das operações da locadora. O lucro total corresponde à soma do valor de todas as fitas que foram devolvidas com sucesso. O lucro total será exibido no formato: “Lucro total R\$%%”, em que %% corresponde ao valor do lucro total.

Seu código deve estar modularizado! Ele deve conter um tipo *tLocadora* e um tipo *tFilme*. O tipo *tLocadora* será responsável por controlar o estoque da locadora, enquanto o tipo *tFilme* será responsável por representar um filme.

O tipo *tLocadora* possuirá, pelo menos, as quatro funções a seguindo o cabeçalho seguir (outras podem ser necessárias para evitar acessos indevidos a seus atributos).

- Função para cadastrar o filme: *tLocadora CadastrarFilmeLocadora(tLocadora locadora, char nome[], int codigo, int valor, int quantidade);*
- Função para alugar o filme: *tLocadora AlugarFilmesLocadora(tLocadora locadora, int codigos[], int quantidadeCodigos);*
- Função para devolver o filme: *tLocadora DevolverFilmesLocadora(tLocadora locadora, int codigos[], int quantidadeCodigos);*
- Função para consultar o estoque: *void ConsultarEstoqueLocadora(tLocadora locadora);*

O tipo *tFilme* também possuirá funções para sua manipulação. O aluno deve pensar nas funções de forma a evitar acessos indevidos aos atributos do tipo (por exemplo, *CriarFilme*, entre outras).

Veja exemplos de formatação das entradas e saídas a seguir.

Exemplo de Entrada:

```
Cadastrar
2,Shrek,4,1
3,A Mumia,4,3
1,Matrix,2,3
6,A Sociedade Do Anel,5,1
1,A Viagem,5,1
#
Devolver
1
#
Alugar
1
2
5
#
Alugar
2
#
Devolver
2
#
Estoque
#
```

Exemplo de Saída:

Filme cadastrado 2 - Shrek
Filme cadastrado 3 - A Mumia
Filme cadastrado 1 - Matrix
Filme cadastrado 6 - A Sociedade Do Anel
Filme ja cadastrado no estoque

Nao e possivel devolver o filme 1 - Matrix.

Filme 5 nao cadastrado.
Total de filmes alugados: 2 com custo de R\$6

Filme 2 - Shrek nao disponivel no estoque. Volte mais tarde.

Filme 2 - Shrek Devolvido!

~ESTOQUE~
3 - A Mumia Fitas em estoque: 3
6 - A Sociedade Do Anel Fitas em estoque: 1
1 - Matrix Fitas em estoque: 2
2 - Shrek Fitas em estoque: 1

Lucro total R\$4

ANEXO I

A correção da questão será feita de forma incremental, onde cada caso de teste apresentado cobra certos conceitos além do caso anterior, permitindo que a nota também seja incremental.

a) Caso de teste 1 - Apenas cadastro de filmes e consulta ao estoque.

Entrada:

```
Cadastrar
2,Shrek,4,1
#
Estoque
#
```

Saída esperada:

```
Filme cadastrado 2 - Shrek

~ESTOQUE~
2 - Shrek Fitas em estoque: 1
```

b) Caso de teste 2 Cadastro de filmes, Aluguel e Consulta ao Estoque

Entrada:

```
Cadastrar
1,A Sociedade Do Anel,5,1
#
Alugar
1
#
Estoque
#
```

Saída esperada:

Filme cadastrado 1 - A Sociedade Do Anel

Total de filmes alugados: 1 com custo de R\$5

~ESTOQUE~

1 - A Sociedade Do Anel Fitas em estoque: 0

c) Caso de teste 3 Cadastro de filmes, Aluguel, Devolução e Consulta ao Estoque

Entrada:

Cadastrar
1,A Sociedade Do Anel,5,1

Alugar
1

Devolver
1

Estoque
#

Saída esperada:

Filme cadastrado 1 - A Sociedade Do Anel
Total de filmes alugados: 1 com custo de R\$5
Filme 1 - A Sociedade Do Anel Devolvido!
~ESTOQUE~
1 - A Sociedade Do Anel Fitas em estoque: 1
Lucro total R\$5

d) Caso de teste 4 - Cadastro de filmes, Aluguel, Erro no Aluguel, Devolução e Consulta ao Estoque

Entrada:

Cadastrar
2,Shrek,4,1
3,A Mumia,4,3

Alugar
1
2
3

Devolver
2

Estoque
#

Saída esperada:

Filme cadastrado 2 - Shrek
Filme cadastrado 3 - A Mumia

Filme 1 nao cadastrado.
Total de filmes alugados: 2 com custo de R\$8

Filme 2 - Shrek Devolvido!

~ESTOQUE~
3 - A Mumia Fitas em estoque: 2
2 - Shrek Fitas em estoque: 1

Lucro total R\$4

e) Caso de teste 4 - Cadastro de filmes, Erro no cadastro de filmes, Aluguel, Erro no Aluguel, Devolução, Erro na Devolução e Consulta ao Estoque

Entrada:

```
Cadastrar
2,Shrek,4,1
3,A Mumia,4,3
1,Matrix,2,3
6,A Sociedade Do Anel,5,1
1,A Viagem,5,1
#
Devolver
1
#
Alugar
1
2
5
#
Alugar
2
#
Devolver
2
#
Estoque
#
```

Saída esperada:

```
Filme cadastrado 2 - Shrek
Filme cadastrado 3 - A Mumia
Filme cadastrado 1 - Matrix
Filme cadastrado 6 - A Sociedade Do Anel
Filme ja cadastrado no estoque

Nao e possivel devolver o filme 1 - Matrix.

Filme 5 nao cadastrado.
Total de filmes alugados: 2 com custo de R$6

Filme 2 - Shrek nao disponivel no estoque. Volte mais tarde.

Filme 2 - Shrek Devolvido!

~ESTOQUE~
3 - A Mumia Fitas em estoque: 3
6 - A Sociedade Do Anel Fitas em estoque: 1
1 - Matrix Fitas em estoque: 2
2 - Shrek Fitas em estoque: 1
```


Lucro total R\$4