



TRABALHO 2

Valor: 6,0 pontos.

Data de entrega: até às 21h do dia 25/11/2019 via SIGAA.

- O trabalho deverá ser realizado, **obrigatoriamente**, em grupos de 5 pessoas.
- O seu grupo pode ser composto de alunos de outras turmas, **desde que todos sejam alunos de turmas práticas da profª. Simone Streitenberger e estejam designados a realizar este trabalho.**
- Você tem até o dia 15/11/2019 (sexta-feira) às 23h59 para **formar os grupos** e informar à professora via e-mail (simonecs@unifei.edu.br). Apenas um dos componentes deverá enviar o e-mail detalhando o nome, RA e turma de **todos os alunos** que comporão seu grupo.
- O aluno que enviar o e-mail informando os componentes do grupo **receberá um retorno, também por e-mail, validando a criação do grupo.**
- Após a entrega do trabalho, qualquer componente do grupo poderá ser convocado a explicar a lógica utilizada na elaboração do programa do seu grupo.

Objetivo do trabalho: avaliar os conceitos aprendidos na disciplina ECO105 – Fundamentos de Lógica de Programação.

Critérios de avaliação:

1. Execução correta do programa. **(2,0 pontos)**
2. Lógica aplicada e adequação do código a tudo o que foi exigido no enunciado (criação e chamadas das funções da forma como foram especificadas, por exemplo). **(2,0 pontos)**
3. Usabilidade: o usuário deverá ser capaz de entender, por meio de mensagens claras, completas e informativas, quais as opções ele pode escolher dentro do programa, quais informações estão sendo solicitadas e o que está sendo apresentado como resultado. As mensagens deverão contemplar caracteres especiais, conforme exigência da língua portuguesa. **(0,75 pontos)**
4. Legibilidade: nomes de variáveis e constantes que sugiram o que elas irão armazenar, indentação do código, etc. **(0,75 pontos)**
5. Comentários: deverão ser inseridos comentários ao longo do programa que auxiliem no entendimento da lógica criada (utilizar o bom senso). **(0,5 pontos)**

ORIENTAÇÕES

- O trabalho deverá seguir todas as **exigências detalhadas neste documento.**
- Na data de entrega do trabalho, **apenas um dos componentes do grupo deverá submeter o trabalho via SIGAA.**
- O arquivo a ser enviado no SIGAA deverá ser somente aquela cuja extensão seja **.cpp**.
- O nome do arquivo deverá ser: trabalho2.cpp
- Dentro do programa, **as primeiras linhas** deverão conter, **obrigatoriamente**, o seguinte cabeçalho inserido como **comentário**:

/* ECO105.2 – Fundamentos de Lógica de Programação (prática) – Trabalho 2

Aluno:	RA:	Turma:
Aluno:	RA:	Turma:
Aluno:	RA:	Turma:
Aluno:	RA:	Turma:
Aluno:	RA:	Turma:

*/

ATENÇÃO: conforme mencionado acima, o trabalho deverá ser realizado em grupos de 5 pessoas. Trabalhos realizados por grupos formados por um número menor ou maior de pessoas não serão avaliados e receberão nota zero. Além disso, TRABALHOS QUE CONTENHAM LÓGICA IGUAL OU MUITO PARECIDA (QUE SUGIRAM CÓPIA) SERÃO INVALIDADOS E RECEBERÃO NOTA ZERO, SEM CHANCES DE NOVA REALIZAÇÃO E SEM CHANCES DE ALTERAÇÃO POSTERIOR DA NOTA.
Não será permitida a entrega do trabalho após o prazo determinado!



Objetivo do programa: controlar o estoque de uma mercearia.

Exigências:

O seu programa deverá conter, **obrigatoriamente**, as seguintes funções. Você pode definir quais os parâmetros cada uma das funções deverá receber e de que forma eles serão passados (por valor ou por referência).

- **int** menu();
- **void** cadastrarProdutos();
- **bool** pesquisarProduto();
- **void** entradaEstoque();
- **void** saidaEstoque();
- **void** relatorioProduto();
- **void** relatorioGeral();

Além disso, o programa deverá contemplar, **obrigatoriamente**, as seguintes estruturas:

```
struct produto
{
    int codigoProduto;
    string descricao;
    char categoria;
};

struct estoque
{
    int codigoProduto;
    int quantidade;
};
```

1. A **função** chamada **menu()** será responsável por apresentar na tela as operações oferecidas (conforme imagem), solicitar ao usuário inserir o número correspondente à ação que ele deseja e **retornar** este valor digitado para que a realização da operação associada possa ser disparada **dentro da função principal**.

```
Menu de opções:
1 - Cadastrar produtos.
2 - Entrada no estoque.
3 - Saída de estoque.
4 - Exibir relatório por produto.
5 - Exibir relatório geral.
6 - Finalizar programa.

Digite o número correspondente a sua opção:
```

- **Enquanto** o número digitado não corresponder a nenhuma das ações especificadas, deverá ser emitida uma mensagem informando **“OPÇÃO INVÁLIDA”**, a tela deverá ser limpa e o **menu reexibido**.



- Se o usuário digitar uma ação válida, a tela deverá ser limpa para que as informações correspondentes à opção escolhida possam ser exibidas.
 - Se o usuário escolher a opção Finalizar programa, o programa deverá **exibir a mensagem “O PROGRAMA ESTÁ SENDO FINALIZADO...”** antes de ser encerrado.
2. A **função** chamada **cadaststrarProdutos()** deverá receber como parâmetros um vetor de **produtos p[]** e um vetor de **estoque c[]**, ambos de tamanho 5.
- Quando o usuário acessar esta função pela primeira vez, ele deverá, obrigatoriamente, cadastrar 5 produtos, que é a quantidade de produtos que este programa irá controlar.
 - Caso ele acesse a função novamente, deverá ser informado através de uma mensagem de que **"O LIMITE DE PRODUTOS CADASTRADOS FOI ATINGIDO!"** e o programa deverá voltar ao menu principal.
 - Para cadastrar um produto, o usuário deverá informar, primeiramente, o **código do produto**. Este código deverá ser pesquisado no vetor de **produtos p[]**.
 - O **código de produto é único**, ou seja:
 - i. **Enquanto** o usuário inserir um código de produto que já exista, ele deverá ser informado de que **“ESTE PRODUTO JÁ ESTÁ CADASTRADO!”**, e solicitado a inserir um novo código de produto.
 - ii. No momento que o usuário inserir um **código de produto válido**, o programa deverá solicitar a **descrição** e a **categoria do produto**. As três informações deverão, então, ser armazenadas em uma posição de índice **i** do vetor de **produtos p[]**.
- Atenção:** antes do comando `getline` para armazenar a descrição do produto, insira a instrução `fflush(stdin);`
- A cada **produto registrado** no vetor de **produtos p[]**, o **código de produto** deverá ser **replicado** para a mesma posição de índice **i** do vetor de **estoque c[]**. O campo **quantidade** correspondente deverá receber valor zero, neste momento.
3. A **função** chamada **pesquisarProduto()** deverá receber como parâmetros um vetor de **produtos p[]** e um código de produto que se deseja pesquisar neste vetor.
- Esta função deverá buscar o código de produto recebido dentro do vetor de **produtos p[]** e **retornar true** se o valor for encontrado ou **false** se o valor não for encontrado.
 - Caso o código do produto seja encontrado, deverá **retornar** também o **índice** do produto dentro do vetor.
4. A **função** chamada **entradaEstoque()** deverá receber como parâmetros um vetor de **produtos p[]** e um vetor de **estoque c[]**, ambos de tamanho 5.
- Caso o usuário acesse esta função, mas não exista nenhum produto cadastrado, o programa deverá emitir uma mensagem **“NÃO HÁ PRODUTOS CADASTRADOS!”** e voltar ao menu principal.
 - O programa só permite que sejam inseridas no estoque quantidades para **produtos já cadastrados**. Desta forma, a primeira informação que o usuário precisa preencher para realizar a entrada no estoque é o **código do produto**. Este código deverá ser pesquisado no vetor de **produtos p[]**.



- i. Caso o usuário tente cadastrar uma quantidade para um produto não cadastrado, o programa deverá emitir uma mensagem **“PRODUTO NÃO ENCONTRADO!”** e voltar ao menu principal.
 - ii. Caso o produto seja válido, deverá ser solicitada a quantidade que se deseja dar entrada e adicioná-la à variável **quantidade** da respectiva posição do vetor de **estoque c[]**.
- 5. A **função** chamada **saidaEstoque()** deverá receber como parâmetros um vetor de **produtos p[]** e um vetor de **estoque c[]**, ambos de tamanho 5.
 - Caso o usuário acesse esta função, mas não exista nenhum produto cadastrado, o programa deverá emitir uma mensagem **“NÃO HÁ PRODUTOS CADASTRADOS!”** e voltar ao menu principal.
 - O programa só permite que sejam informadas saídas de estoque para **produtos já cadastrados**. Desta forma, a primeira informação que o usuário precisa preencher para realizar a retirada de estoque é o **código do produto**. Este código deverá ser pesquisado no vetor de **produtos p[]**.
 - i. Caso o usuário tente informar uma saída para um produto não cadastrado, o programa deverá emitir uma mensagem **“PRODUTO NÃO ENCONTRADO!”** e voltar ao menu principal.
 - ii. Caso o produto seja válido, deverá ser solicitada a quantidade que se deseja dar baixa em estoque.
 - a) Caso o produto não possua quantidade suficiente para a retirada (inferior ao que o usuário está tentando dar baixa), deverão ser emitidas as seguintes mensagens **“NÃO É POSSÍVEL FAZER ESTA RETIRADA! A QUANTIDADE DISPONÍVEL É DE: xxx. REAVALIE A RETIRADA E REPITA A OPERAÇÃO.”**.
 - b) Caso o produto possua quantidade suficiente para a retirada, a operação é realizada atualizando a variável **quantidade** da respectiva posição do vetor de **estoque c[]**.
- 6. A **função** chamada **relatorioProduto()** deverá receber como parâmetros um vetor de **produtos p[]** e um vetor de **estoque c[]**, ambos de tamanho 5.
 - Caso o usuário acesse esta função, mas não exista nenhum produto cadastrado, o programa deverá emitir uma mensagem **“NÃO HÁ PRODUTOS CADASTRADOS!”** e voltar ao menu principal.
 - O programa só permite que seja emitido um relatório de estoque para **produtos já cadastrados**. Desta forma, a primeira informação que o usuário precisa preencher para solicitar o relatório é o **código do produto**. Este código deverá ser pesquisado no vetor de **produtos p[]**.
 - i. Caso o usuário tente emitir relatório para um produto não cadastrado, o programa deverá exibir uma mensagem **“PRODUTO NÃO ENCONTRADO!”** e voltar ao menu principal.
 - ii. Caso o produto seja válido, o programa deverá emitir o relatório contendo as informações de código do produto, descrição, categoria e quantidade em estoque (conforme imagem).



```
Insira o código do produto: 123
PRODUTO: 123
Descrição: Caderno Tilibra 100 folhas
Categoria: A
Quantidade disponível em estoque: 45
Pressione qualquer tecla para continuar. . .
```

7. A função chamada **relatorioGeral()** deverá receber como parâmetros um vetor de **produtos p[]** e um vetor de **estoque c[]**, ambos de tamanho 5.
- Caso o usuário acesse esta função, mas não exista nenhum produto cadastrado, o programa deverá emitir uma mensagem **“NÃO HÁ PRODUTOS CADASTRADOS!”** e voltar ao menu principal.
 - Caso existam produtos cadastrados, o programa deverá apresentar o relatório de **todos os produtos**, seguindo o mesmo padrão de informações apresentadas no relatório individual (conforme imagem).

```
*** RELATÓRIO GERAL ***
PRODUTO: 123
Descrição: Caderno Tilibra 100 folhas
Categoria: A
Quantidade disponível em estoque: 45

PRODUTO: 74
Descrição: Caneta BIC Azul
Categoria: A
Quantidade disponível em estoque: 0

PRODUTO: 362
Descrição: Chocolate ao leite Nestle 100gr
Categoria: B
Quantidade disponível em estoque: 17

PRODUTO: 984
Descrição: Chocolate BIS branco caixa
Categoria: B
Quantidade disponível em estoque: 0

PRODUTO: 7
Descrição: Pasta de dente Colgate 90gr
Categoria: C
Quantidade disponível em estoque: 40
Pressione qualquer tecla para continuar. . .
```