UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
CURSO DE CIÊNCIA DA COMPUTAÇÃO
PROGRAMAÇÃO DE COMPUTADORES

TRABALHO PRÁTICO 3

Joãozinho decidiu abrir seu próprio negócio. Como ele está terminando o curso de programação, decidiu ele mesmo implementar o sistema de vendas e controle de estoque da sua empresa. Seu objetivo é ser um fornecedor de matéria prima para as várias pizzarias da cidade.

Seu sistema funcionará da seguinte forma:

- Cada cliente vai enviar seu pedido através de um arquivo texto contendo os produtos e quantidades desejadas, conforme exemplo abaixo:

```
Pizzaria Mamute
---------------
massa 20
queijo 5
Massa 10
frango 5
molho 40
Queijo 30
FRANGO 10
presunto 18
```

- O programa deve ler o conteúdo do pedido e gerar um recibo no formato abaixo:

```
Pizzaria Mamute
---------------------------------------
Massa: 30kg a R$2.50/kg = R$75.00
Queijo: 35kg a R$15.00/kg = R$525.00
Frango: 15kg a R$9.00/kg = R$135.00
Molho: 40kg a R$8.20/kg = R$328.00
Presunto: 18kg a R$10.00/kg = R$180.00
---------------------------------------
Compra = R$1243.00
Desconto = R$124.30
Total = R$1118.70
```

Observe que no pedido os nomes podem usar letras maiúsculas ou minúsculas e podem haver pedidos do mesmo produto em linhas diferentes. Além disso, a empresa dá um desconto de `10%` para compras de valor igual ou superior a `R$1000.00.`

Uma vez gerado o recibo de compra, a quantidade em estoque de cada produto deve ser atualizada no sistema. O sistema deve ser composto por um menu de opções que é exibido na tela até que seja escolhida a opção (S) de Sair.

```
Sistema de Controle
===================
(P)edir
(A)dicionar
(E)xcluir
(L)istar
(S)air
===================
Opção: [_]
```

Cada opção deve permitir que o usuário entre com os dados necessários para completar aquela tarefa. As opções são acionadas pelo pressionamento da primeira letra (maiúscula ou minúscula) de cada opção. Abaixo está uma descrição para cada uma delas:

- **Pedir**: deve ler um arquivo de pedido, gravar um recibo para o pedido e atualizar as quantidades dos produtos em estoque. O recibo é um arquivo texto com o mesmo nome do arquivo de pedido, porém com a extensão ".nfc" (ex.: mamute.nfc). Caso algum dos produtos solicitados não estejam em estoque, ou estejam em quantidade insuficiente, o pedido deve falhar, exibindo uma mensagem do porque o pedido não pôde ser atendido. Em caso de falha, o recibo não deve ser gerado e o estoque não deve ser alterado para nenhum produto.

  ```
  Pedir
  -----
  Arquivo: mamute.txt

  Pedido falhou!
  Molho: Solicitado = 40kg / Em estoque = 30kg
  Massa: Solicitado = 30kg / Em estoque = 10kg
  ```

- **Adicionar**: deve adicionar um novo registro ao vetor de estoque. Este registro deve conter o nome do produto, o preço e a quantidade do produto que está chegando ao estoque. Caso o produto já exista, deve-se apenas atualizar o preço e acrescentar a quantidade informada ao valor atual do estoque.

  ```
  Adicionar
  ---------
  Produto: Massa
  Preço: 2.50
  Quantidade: 50
  ```

- **Excluir**: deve mostrar uma lista numerada (a partir de 1) dos produtos existentes no estoque. Quando um número for escolhido, o programa deve, após confirmar a exclusão, remover o registro do produto do vetor de estoque.

  ```
  Excluir
  -------
  1) Massa
  2) Queijo
  3) Molho

  Número do produto: 1
  Deseja excluir "Massa" (S/N)? s
  ```

- **Listar**: deve mostrar a lista de produtos existentes no vetor de estoque, com seu nome, valor unitário e quantidade.

```
Listagem
--------
Massa  - R$2.50  - 10kg
Queijo - R$15.00 - 100kg
Molho  - R$8.20  - 30kg
```

EXIGÊNCIAS

Use um **arquivo binário** para guardar as informações dos produtos em estoque. Este arquivo deve ser lido na entrada do programa e seus dados passados para um vetor dinâmico. O arquivo deve ser atualizado **apenas ao final do programa**. Quando a opção (S) Sair for selecionada, o programa deve sobrescrever o arquivo binário com o conteúdo atualizado do vetor dinâmico. Enquanto o programa estiver rodando, adições, exclusões, listagens e processamentos de pedidos devem ser feitas sobre os dados do vetor dinâmico.

Caso o arquivo de estoque ainda não exista, o programa deve simplesmente iniciar com o vetor dinâmico vazio. Não há necessidade de criar o arquivo na entrada do programa, pois na saída o arquivo deve ser aberto para escrita, sendo então criado nesse momento.

O **vetor dinâmico** deve ter seu tamanho inicializado para a quantidade de produtos no arquivo. Se a capacidade do vetor for excedida durante a execução do programa, um novo vetor deve ser criado com a nova capacidade igual a capacidade anterior + 2^n, sendo n a quantidade de vezes que o vetor foi expandido. Os dados devem ser copiados do vetor antigo para o novo e o vetor antigo deve ser destruído. Abstraia o processo de expansão para o resto do programa mantendo **um ponteiro** sempre apontando para o vetor mais recente. O programa deve sempre usar esse ponteiro para acessar o conteúdo do vetor.

ENTREGA DO TRABALHO

Grupos: Trabalho individual

Data da entrega: 07/02/2020 (até a meia noite)

Valor do Trabalho: 3,0 pontos (na 3a Unidade)

Forma de entrega: enviar apenas os arquivos fonte (.cpp) e os arquivos de inclusão (.h) compactados no formato **zip** através da tarefa correspondente no SIGAA.

O não cumprimento das orientações resultará em **penalidades:**

- Programa não executa no Visual Studio 2019 (3,0 pontos)
- Programa contém partes de outros trabalhos (3,0 pontos)
- Atraso na entrega (1,5 pontos por dia de atraso)
- Arquivo compactado em outro formato que não zip (0,5 ponto)
- Envio de outros arquivos que não sejam os .cpp e .h (0,5 ponto)
- Programa sem comentários e/ou desorganizado (0,5 ponto)