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

TRABALHO PRÁTICO 3

Depois de embarcar na onda da IA e criar seu próprio ChatBot, Joãozinho se deu conta que essa coisa de IA pode muito bem ser uma grande bolha. Ele então decidiu atacar outro ramo de negócios. Seu objetivo é vender salgadinhos para as várias lanchonetes da cidade. Como ele está terminando o curso de programação, decidiu ele mesmo implementar o sistema de vendas e controle de estoque da sua empresa.

Seu sistema funcionará da seguinte forma:

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

```
Super Lanches
-------------
esfirras 20
COXINHA 15
Empadas 10
Kibe 14
Risole 10
empada 5
Esfirra 10
```

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

```
Super Lanches
----------------------------------------
Esfirra    30   2.50/und.  =   R$75.00
Coxinha    15   3.00/und.  =   R$45.00
Empada     15   1.50/und.  =   R$22.50
Kibe       14   2.00/und.  =   R$28.00
Risole     10   1.00/und.  =   R$10.00
----------------------------------------
                  Compra   =   R$180.50
                Desconto   =   R$18.05
                   Total   =   R$162.45
```

Observe que no pedido os nomes dos salgadinhos podem usar letras maiúsculas ou minúsculas, podem estar no singular ou no plural e pode 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$100.00`.

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) para sair.

```
Salgadinhos & Cia
==================
[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.: SuperLanches.nfc). Caso algum dos produtos solicitados não esteja em estoque, ou esteja em quantidade insuficiente, o pedido deve falhar, exibindo uma mensagem do porquê 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: SuperLanches.txt

  Pedido falhou!
  Empada: Solicitado = 15 und. / Em estoque = 12 und.
  Kibe: Solicitado = 14 und. / Em estoque = 1 und.
  ```

- **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: Kibe
  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 vetor de estoque. O programa não deve reduzir o tamanho do vetor, apenas considerar que o número de elementos válidos no vetor foi reduzido em uma unidade. Novas inserções irão aproveitar o espaço existente no vetor.

```
Excluir
-------
1) Coxinha
2) Kibe
3) Empada

Número do produto: 2
Deseja excluir "Kibe" (S/N)? s
```

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

```
Listagem
--------
Coxinha  - R$3.00 -  10 und.
Empada   - R$1.50 - 100 und.
Esfirra  - R$2.50 -  50 und.
```

EXIGÊNCIAS

1) O programa deve ser construído apenas com os assuntos vistos durante o curso, entretanto os seguintes assuntos, que foram abordados na disciplina, não devem ser usados no trabalho:

    a. Não utilize variáveis globais
    b. Não utilize a biblioteca <string>
    c. Não utilize a biblioteca <vector> ou <array>

2) 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 feitos sobre o vetor dinâmico.

3) Caso o arquivo de estoque não exista, o programa deve simplesmente iniciar com o vetor dinâmico vazio, ou seja, o ponteiro do vetor dinâmico deve apontar para nulo (nullptr). Não há necessidade de criar o arquivo binário na entrada do programa, pois na saída o arquivo deve ser aberto para escrita, sendo então criado neste momento.

4) 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 liberado da memória. Abstraia o processo de expansão para o resto do programa fazendo com que sempre exista **um ponteiro** 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: 10/12/2025 (até a meia noite)

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

Forma de entrega:

Ativar a exibição de arquivos ocultos no explorador de arquivos, apagar a pasta oculta .vs da pasta da solução, apagar a pasta x64 da pasta da solução, apagar a pasta x64 da pasta do projeto, apagar arquivo com a extensão .suo, se ele existir em qualquer uma das pastas.

Após as exclusões, confira se os arquivos restantes, que estão na pasta da solução e na pasta do projeto, têm uma das extensões a seguir. Nenhum outro tipo de arquivo deve ser enviado.

- .h – arquivo de cabeçalho
- .cpp – arquivo de código fonte
- .sln – solução do visual studio
- .slnx – solução do visual studio
- .vcxproj – projeto do visual studio
- .vcxproj.filters – filtros do projeto do visual studio
- .vcxproj.user – configurações do projeto do visual studio

Em seguida compactar todo o conteúdo da pasta da solução em um **arquivo .zip** (não usar .rar, .7zip ou outros formatos de compressão) e enviar através da tarefa correspondente no SIGAA.

Penalidades:

O não cumprimento das orientações acima resultarão em descontos na pontuação do trabalho:

- Programa não executa no Visual Studio 2022 (3,0 pontos)
- Programa contém partes de outros trabalhos (3,0 pontos)
- Programa utiliza recursos não vistos na disciplina (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 ou pastas que não sejam os descritos acima (0,5 ponto)
- Programa sem comentários (0,5 ponto)
- Código sem indentação e/ou desorganizado (0,5 ponto)
- Nomes de variáveis e funções sem significado (0,5 ponto)