

4º LABORATÓRIO de Linguagem de Programação

Profa. Eliane

Objetivo: Trabalhar com leitura e escrita de arquivos e com manipulação de strings.

Tarefa:

A empresa Bosh é líder mundial no fornecimento de tecnologias e serviços. Suas operações são divididas em quatro setores de negócios: Tecnologia Automotiva, Tecnologia Industrial, Bens de Consumo, Energia e Tecnologia Predial. O Grupo Bosch é composto pela Robert Bosch GmbH e suas 360 subsidiárias e empresas regionais em aproximadamente 50 países. Incluindo os representantes de vendas e serviços, a Bosch está presente em cerca de 150 países. É uma rede mundial de desenvolvimento, produção e distribuição, com crescimento contínuo. Saiba mais em www.brasil.bosch.com.br

O Grupo Bosch possui 15 divisões de serviços e a tarefa deste laboratório consiste em sistematizar o processo de compra e venda de produtos.

O arquivo pode ter qualquer nome de cliente, fornecedor e produto. Os nomes das peças e fornecedores são nomes usados pela empresa, extraídos das informações públicas disponíveis no site acima referenciado, e servem apenas para apresentar exemplos reais. Sobretudo, a relação fornecedor, peça e clientes são hipotéticos. Portanto, não vamos fazer nada específico para a Bosch!

Requisitos:

1. Produtos, clientes e fornecedores devem ser identificados por seus códigos exclusivos. Abaixo a descrição de clientes e fornecedores. Outros poderão ser acrescentados.

Cada cliente também possui um código no sistema, por exemplo:

- “AUD” para montadora Audi
- “BMW” para montadora BMW
- “CHR” para montadora Chrysler
- “CHE” para montadora Chevrolet
- “HON” para montadora Honda
- “VOL” para montadora Volkswagen

Cada fornecedor também possui um código no sistema, por exemplo:

- “F01SAMOT” para Peças Usinadas - Indústria Mecânica Samot Ltda
- “F02SULBRA” para Peças Plásticas - Sulbras Moldes e Plásticos Ltda
- “F03WETZEL” para Peças Injetadas em Alumínio - Wetzel S/A
- “F04TESSIN” para Peças Estampadas - Tessin Indústria e Comércio Ltda “HON”

para Destaque - São Marco Indústria e Comércio Ltda.

- “F05ZANET” para Sub-Conjuntos Zanettini Barossi S.A Indústria e Comércio Ltda.

Cada peça comercializada também tem um código no sistema, por exemplo:

- *Peças de Motores de arranque para veículos leve*

- a) “MOT51201” para *Starter Pinion*
- b) “MOT51202” para *Drive and Shield*
- c) “MOT51203” para *Multiplate Clutch*
- d) “MOT51204” para *Armature*
- e) “MOT51205” para *Stator Frame*
- f) “MOT51206” para *Excitation Winding*
- g) “MOT51207” para *Commutator Set*
- h) “MOT51208” para *Solenoid Switch*
- i) “MOT51209” para *Control Relay*

- *Peças para alternadores*

- a) “ALT01612001” para *Pulley*
- b) “ALT01612002” para *Roller bearing*
- c) “ALT01612003” para *Stator*
- d) “ALT01612004” para *Regulator*
- e) “ALT01612005” para *Protective cap*
- f) “ALT01612006” para *Rectifier*
- g) “ALT01612007” para *Collector-ring end shield*

2. Considere que teremos, no máximo, 30 clientes, 30 fornecedores e 30 produtos diferentes.
3. Considere que o código de cliente/fornecedor/produto é composto por uma só palavra (sem espaço) com no máximo 10 caracteres. Para facilitar defina um tipo de dados específico e use quando precisar:

```
#typedef char str[11]; //Definir 11 por causa do \0 final
```

4. Existe um conjunto de arquivos que servirá de entrada para o programa:

- CLIENTE.TXT
- CATEGORIA.TXT
- FORNECEDOR.TXT
- PRODUTO.TXT
- ENTRADA.TXT

5. O arquivo “cliente.txt” com a lista ordenada pelo nome do cliente. O arquivo possui as seguintes informações, nesta ordem, separadas por espaço: código do cliente, nome do cliente. Exemplo:

- 6.

AUD Audi
VOL Volkswagen
CHE Chevrolet
BMW BMW
HON Honda
CHR Chrysler

Criar o arquivo chamado categoria.txt com a lista ordenada por nome da categoria. O arquivo possui as seguintes informações, nesta ordem, separadas por espaço: código e nome da categoria.

```
MOT512 Pecas de Motores de arranque para veiculos leves
ALT612 Pecas para alternadores
```

7. Crie o arquivo chamado “fornecedor.txt” com a lista ordenada pelo nome do fornecedor. O arquivo possui as seguintes informações, nessa ordem separadas por espaço: código do fornecedor e nome do fornecedor.

```
F01SAMOT Peças Usinadas - Indústria Mecânica Samot Ltda
F02SULBRA Peças Plásticas - Sulbras Moldes e Plásticos Ltda
F03WETZEL Peças Injetadas em Alumínio - Wetzel S/A
F04TESSIN Peças Estampadas - Tessin Indústria e Comércio Ltda
F05SMARCO para Destaque - São Marco Indústria e Comércio Ltda.
F06ZANET Sub-Conjuntos Zanettini Barossi S.A Indústria e Comércio Ltda.
```

8. Crie o arquivo chamado “produto.txt” com a lista ordenada, por valor unitário. O arquivo possui as seguintes informações, nesta ordem, separadas por espaço: código do fornecedor, código do produto, descrição do produto, valor unitário (valor real, em moeda brasileira), quantidade em estoque (valor inteiro), quantidade mínima no estoque (valor inteiro). No início do arquivo há a quantidade de transações contidas nele. Veja um exemplo:

```
23
F01SAMOT MOT51201 Starter Pinion 45.5 200 150
F02SULBRA MOT51202 Drive and Shield 60.80 180 150
F02SULBRA MOT51203 Multiplate Clutch 37.65 200 150
F02SULBRA MOT51204 Armature 28.99 150 150
F04TESSIN MOT51205 Stator Frame 19.90 190 150
F05SMARCO MOT51206 Excitation Winding 30.50 200 150
F05SMARCO MOT51207 Commutator Set 42.00 210 150
F01SAMOT MOT51208 Solenoid Switch 34.99 220 150
F06ZANET MOT51209 Control Relay 44.00 100 150
F02SULBRA ALT612001 Pulley 35.50 220 200
F04TESSIN ALT612002 Roller bearing 32.80 400 200
F06ZANET ALT612003 Stator 60.59 250 200
F03WETZEL ALT612004 Regulator 26.80 300 200
F06ZANET ALT612005 Protective cap 15.98 300 200
F03WETZEL ALT612006 Rectifier 27,90 100 200
F06ZANET ALT612007 Collector-ring end shield 49.59 190 200
```

Obs: se houver empate de valores, o critério de desempate é a ordem alfabética do nome do produto. Usar **strcmpi** nas comparações de strings.

9. Com base no arquivo `produto.txt`, gerar o arquivo `produto_fornecedor.txt` de acordo com as informações, nesta ordem, separadas por espaço: código do fornecedor, código do produto, descrição do produto, ordenados por código do produto.

```
F01SAMOT MOT51201 Starter Pinion
F01SAMOT MOT51208 Selenoid Switch
```

Obs.:

- a) Cada produto tem apenas um único fornecedor.
- b) Uma categoria de produto possui vários produtos.

10. Considere que as transações de vendas realizadas no mês estão disponibilizadas no arquivo “`entrada.txt`”, localizado no mesmo diretório que o seu `.exe`. O arquivo possui as seguintes informações, nesta ordem separadas por espaço: código do cliente, código do produto, quantidade (valor inteiro), valor unitário. No início do arquivo há a quantidade de transações contidas nele. Veja um exemplo:

```
4
AUD MOT51201 100 58.5
BMW ALT612001 250 45.8
CHR MOT51201 30 68.5
AUD MOT51204 450 56.5
```

- Obs:
- a) A mesma peça pode ser vendida a preços diferentes para clientes diferentes.
 - b) O mesmo cliente pode aparecer mais de uma vez no arquivo com peças diferentes.

11. Ao processar o arquivo `entrada.txt`, o programa deverá debitar a quantidade vendida do estoque de cada peça no arquivo `produto.txt`.

12. Após o processamento do arquivo `entrada.txt`, se a quantidade vendida for maior que a quantidade disponível no estoque OU se a quantidade disponível no estoque após a venda for menor que o estoque mínimo, o programa deverá gerar o arquivo `compras.txt` contendo a relação de produtos a serem comprados pela Bosch, respeitando os seguintes critérios

- a) gerar o arquivo chamado “`compras.txt`” com a lista ordenada, por fornecedor, contendo as seguintes informações: código do fornecedor, código do produto, quantidade. Considere que a quantidade deve atender a necessidade de entrega ao cliente além da reposição do estoque mínimo.

```
F01SAMOT MOT51201 45.5
F03WETZEL MOT51204 450
```

- b) no exemplo, o arquivo `MOT51204` tem 150 unidades no estoque e o estoque mínimo é de 150 unidades. No arquivo `entrada.txt` há um pedido de 450 unidades desta peça.

Como há 150 no estoque, é necessário comprar 300 unidades para atender o pedido e mais 150 unidades para manter o estoque mínimo, resultando num pedido de compras de 450 unidades.

c) Usar **strcmp** nas comparações de strings.

13. Gerar o arquivo chamado “relatório_gerencial.txt” contendo:

- a) Lista ordenada das empresas que mais compraram no mês, em termos de volume financeiro.
- b) Lista ordenada das peças mais vendidas, em termos de quantidade.

14. Gerar o arquivo produtos_por_categoria.txt” contendo a lista ordenada de produtos, agrupados pela categoria.

15. Escolha como deseja processar as informações:

- a. Ler o arquivo de entrada e ir processando conjuntamente; ou
- b. Ler todo arquivo de entrada, guardar as informações em vetores auxiliares e depois processar.

16. Escolha as estruturas auxiliares que você precisa para armazenar os dados processados.

17. Todos os arquivos a serem gerados deverão ser gravados na própria pasta onde está o .exe.

18. Entregar os arquivos relacionados no item 4, usados como entrada no sistema. Os exemplos apresentados servem como base para testes.

19. Utilize boas práticas de programação: indentação, comentários relevantes, variáveis bem definidas, organização em funções, sem goto, sem break, ponto único de return nas funções, sem variáveis globais (exceto vetores e suas dimensões), definição de constantes úteis (*para evitar aparecer o valor numérico repetidamente no código. Ex: dimensão máxima de um vetor*).

Entregar no Teams o códigos fonte e no OneDrive o executável. Esses arquivos deverão ter o nome do aluno e o número do laboratório Lab IV.

Prazo: 3 semanas