

3ª Lista de Exercícios Linguagem de Programação I

Turma	Data de Entrega
LP- Noturno	17/06/2018

Natureza do Trabalho: em dupla

- Entregar a lista através do e-mail palomar.cris@gmail.com até às 23:59 do dia 17/06. Enviar apenas os programas fonte (.c). Os nomes dos arquivos fontes deverão ter no máximo 8 caracteres (além da extensão) e não poderão conter caracteres especiais ou acentuação.
- Cada programa deverá conter um comentário com o Ra e nome do(s) aluno(s).
- Não serão aceitas listas entregues fora do prazo determinado
- Informar se compilado no Cygwin ou CodeBlocks
- A lista merecerá uma nota de 0 a 10, sendo o critério de avaliação definido pelo professor.
- Exercícios que sejam considerados como fruto de algum tipo de fraude por parte da equipe (como cópia total ou parcial, por exemplo) atribuirão nota zero na atividade para TODOS os envolvidos.
- IDENTAR todos os programas.

EXERCÍCIOS

1. Crie um programa que leia um arquivo texto contendo informações de faturas. O programa deverá validar o arquivo, segundo critérios pré-determinados, gerar uma listagem dos erros e mostrar em tela as faturas que estão com os dados corretos. Ao final, o programa deverá exibir a soma de faturas por fornecedor.

Layout do arquivo:

Campo	Posição	Observação
Código Fatura	1 a 4	4 dígitos
Código do Produto	5 a 7	3 dígitos
Nome do Produto	8 a 40	33 dígitos
Quantidade	41 a 44	4 dígitos
Valor Unitário	45 a 54	10 dígitos, sendo os dois últimos correspondentes aos centavos.
Frete	55 a 64	10 dígitos, sendo os dois últimos correspondentes aos centavos.
Fornecedor	65 a 68	4 dígitos

- a) (2,5) Criar uma função para leitura e validação: ler o arquivo fatura.txt (de acordo com o layout acima). Os registros corretos serão inseridos numa matriz de struct. Os registros inválidos serão armazenados no arquivo texto "ErroFatura.txt.

→ Considera-se registro inválido aquele cujo fornecedor ou valor unitário seja igual a zero. Se num mesmo registro, o valor e a quantidade forem iguais a zero, gerar duas linhas no arquivo de erro, uma para cada erro. No arquivo de erro deve ser copiada a linha inteira (no mesmo formato que foi lido) e ao final acrescenta-se o texto "QUANTIDADE DEVE SER MAIOR QUE ZERO" ou "FORNECEDOR DEVE SER INFORMADO"-> **A cada execução do programa, o arquivo de erro deverá ser reinicializado.**

Abaixo, segue exemplo do arquivo de erro:

```
1234100MOCHILA      0000000000100000000010000000 FORNECEDOR DEVE SER INFORMADO
1234100MOCHILA      0000000000100000000010000000 QUANTIDADE DEVE SER MAIOR QUE ZERO
1234101CADERNO      0000000000110000000010001002 QUANTIDADE DEVE SER MAIOR QUE ZERO
1111100MOCHILA      0000000000158000000010001003 QUANTIDADE DEVE SER MAIOR QUE ZERO
1234100MOCHILA      0013000000100000000010000000 FORNECEDOR DEVE SER INFORMADO
```

- b) (1,0) Criar uma função que exiba em tela as faturas corretas através da matriz de struct, em ordem de número de fatura.

Fatura	Produto	Qtde	Unitario	Frete	Total	Fornecedor
1111	103 SULFITE 100 FLS	8	16.70	10.00	143.60	1003
1111	105 GIZ DE CERA	10	18.00	10.00	190.00	1003
1112	100 MOCHILA	9	17.00	10.00	163.00	1003
1234	102 LAPISEIRA	5	13.00	10.00	75.00	1002
1234	103 SULFITE 100 FLS	6	14.90	10.00	99.40	1003
1235	100 MOCHILA	4	12.00	10.00	58.00	1001
4455	100 MOCHILA	11	19.00	10.00	219.00	1001
4455	105 GIZ DE CERA	12	20.00	10.00	250.00	1001

- c) (1,5) Criar uma função que exiba o fornecedor e a soma do total de seus respectivos produtos nas faturas.

⇒ Deverá ser utilizada a função SUBSTRING criada anteriormente em aula para "quebrar" a linha lida do arquivo e separá-la de acordo com o layout do arquivo.

2. (2,5) Neste problema sua tarefa será ler vários números e em seguida dizer quantas vezes cada número aparece na entrada de dados, ou seja, deve-se escrever cada um dos valores distintos que aparecem na entrada por ordem crescente de valor. **UTILIZAR PESQUISA BINÁRIA**

Entrada

A entrada contém apenas 1 caso de teste. A primeira linha de entrada contém um único inteiro N , que indica a quantidade de valores que serão lidos para X ($1 \leq X \leq 2000$) logo em seguida. Com certeza cada número não aparecerá mais do que 20 vezes na entrada de dados.

Saída

Imprima a saída de acordo com o exemplo fornecido abaixo, indicando quantas vezes cada um deles aparece na entrada por ordem crescente de valor.

Exemplo de Entrada	Exemplo de Saída			
7	4	aparece	1	vez(es)
8	8	aparece	2	vez(es)
10	10	aparece	3	vez(es)
8	260	aparece	1	vez(es)
260				
4				
10				
10				

3. (2,5) Verifique se uma palavra é palíndromo usando **recursividade** (Ex. aba, abcba, xyzyx). Para isto crie **uma função** que retorne o número 1 se a palavra recebida for palíndromo e 0 senão for. O programa deve solicitar várias palavras para análise até que a string FIM seja digitada.