

Universidade Federal Fluminense

Instituto de Ciência e Tecnologia – Departamento de Computação

Lista de exercícios - Programação com Arquivos

Arquivos texto

1- Faça um programa que analise a quantidade de vezes que cada uma das letras maiúsculas e minúsculas ou cada um dos algarismos de 0 a 9 aparece em um arquivo texto. O programa deve perguntar o nome do arquivo a ser lido e mostrar o resultado em ordem decrescente de acordo com a quantidade de aparições.

2- Faça um programa que leia o arquivo texto “valores.txt” que está no formato CSV (comma-separated values). Nesse formato, cada linha possui diversos valores (inteiros, reais ou strings) que estão separadas uns dos outros apenas por uma vírgula (não há espaços em branco entre os valores; espaços em branco só existem entre duas palavras da mesma string). O programa deve mostrar quantos números inteiros, reais e strings existem em cada linha do arquivo. Um número só será considerado como real se possuir “.” (ex: 5.478). Um número só será considerado inteiro se possuir apenas algarismos de “0..9”. Qualquer outro tipo de informação é considerado string.

Arquivos binários

1- Faça um programa no qual o usuário possa registrar os dias e horários das suas reuniões. O programa deve ter um menu com duas opções: 1- Registrar uma reunião; 2- Mostrar reuniões agendadas. Ao registrar uma nova reunião, o programa deve perguntar a data (dia, mês e ano), o horário (hora e minuto) e o assunto da reunião (string com 20 caracteres) e escrever no final de um arquivo binário (“agenda.bin”). Ao consultar as reuniões agendadas, o usuário deve informar a data desejada. O programa então vai abrir o arquivo e mostrar na tela os horários e os assuntos das reuniões agendadas na data especificada.

2- Dois arquivos binários “arq1.bin” e “arq2.bin” contém diversos números reais. Faça um programa que leia esses números de uma única vez e os armazene em dois vetores dinâmicos v1 e v2. A seguir, o programa deve procurar se os números de v2 aparecem na mesma ordem em v1. O programa deve escrever no arquivo “resultado.bin” primeiro todos os números de v1, depois todos os números de v2 e por fim em qual posição de v1 começou a coincidência (número inteiro do índice de v1). Se os números de v2 não aparecem na mesma ordem, o valor a ser escrito no final do arquivo será -1. Veja o exemplo abaixo:

V1

4.0 9.1 6.5 5.4 4.2 3.1 7.0 8.2 3.9 0.6 7.3

V2

4.2 3.1 7.0 8.2

Resultado.bin

4.0 9.1 6.5 5.4 4.2 3.1 7.0 8.2 3.9 0.6 7.3 4.2 3.1 7.0 8.2 4

### Classificação Externa

- 1- Os algoritmos de classificação externa são compostos de duas fases. Explique cada uma delas.
- 2- Explique a diferença entre a Intercalação Balanceada de N Caminhos e a Intercalação Ótima.
- 3- De acordo com o algoritmo de classificação (particionamento) chamado de Seleção Natural, mostre como ficam o reservatório e as partições, no momento em que elas forem fechadas (encerradas). Utilize o arquivo abaixo como exemplo e considere o tamanho de registros em memória M=4.

31	28	4	9	8	12	19	20	21	5
24	15	13	10	39	35	22	20	27	3
11	1	14	33	34	29	21	20	26	2
7	17	27	37	47	42	31	10	55	45

### Estruturação de arquivos binários

1- Responda:

- a) O que significa dizer que um campo é chave primária de um registro?
- b) Quais tipos de chaves existem?
- c) Explique com a busca binária pode ser mais eficiente do que a busca linear.

2- Suponha que um aluno esteja fazendo um jogo de gerenciamento de times de futebol. A ficha de cada jogador deve possuir os atributos principais do jogador e informações da passagem desse jogador por outros clubes anteriormente. Proponha um struct para cada um desses dois tipos de informação e crie arquivos para pelo menos 3 jogadores com passagens por pelo menos 4 clubes cada.

3- O arquivo binário "original.bin" representa uma base de dados consolidados sobre reuniões que foram marcadas por uma empresa, que atualmente está sob auditoria interna. O arquivo texto "correcoes.txt" é um arquivo de transações para o modelo chamado de Balance-Line. O arquivo "original.bin" é composto por diversos registros modelados pelo struct reuniao abaixo. O arquivo "correcoes.txt" possui os dados (em cada linha) de uma alteração a ser realizada nos dados de

“original.bin”. Ambos os arquivos já estão ordenado pela chave principal “id”. Implemente o algoritmo Balance-Line (veja a vídeo-aula com atenção) que realiza as operações descritas em “correcoes.txt” e gere um arquivo “corrigido.bin” com as informações atualizadas. Alterações inconsistentes devem ser mostradas na tela. OBS: no arquivo de transações, I significa inclusão de um novo registro (os dados estão todos na mesma linha do arquivo texto), M significa modificação do valor de um campo (o nome do campo e o novo valor estão na mesma linha) e E significa a exclusão completa de um registro.

Dica: leia apenas 1 registro de cada vez, tanto do arquivo texto (linha), quanto do binário (struct reunião); em seguida, compare os respectivos Ids e a operação indicada na linha. Essa comparação vai definir o que deve ser feito com esses registros. Realize a tarefa adequada e só então leia um novo registro (do arquivo texto, do binário ou de ambos, dependendo do caso). NÃO TENTEM LER TUDO DE UMA VEZ SÓ! O programa vai acabar quando todos os registros tiverem sido lidos de ambos os arquivos.

```
struct reuniao{
    int id;
    int dia,mes,ano;
    int hora,minuto;
    int qtdPessoas;
};
```

4- Mostre como ficam os arquivos em cada etapa do algoritmo de Lefkowitz (indexação multi-listas) para as chaves secundárias “estado civil” e “cidade”, da base de dados abaixo. ATENÇÃO: o arquivo não está ordenado pela chave primária. Dessa forma, ele deve terminar com a mesma ordem inicial dos registros, obviamente acrescidos com as colunas incorporadas pelo algoritmo. Utilizem o ED para realizar as ordenações.

Cliente (chave prim.)	Nome (não-chave)	Estado Civil	Cidade
23	João Gonçalves	Casado(a)	Gramado
41	Nair Rabelo	Solteiro(a)	Limeira
71	José Simeão	Viúvo(a)	Goiânia
12	Antonia Sofia	Divorciado(a)	Rio de Janeiro
18	Raquel Neves	Casado(a)	Natal
91	Felipe Fernandes	Casado(a)	Salvador
55	Helio Gutierrez	Viúvo(a)	Limeira
11	Frederico Gomes	Solteiro(a)	Rio de Janeiro
92	Sônia Ferreira	Solteiro(a)	Gramado
31	Tadeu Cabral	Casado(a)	Goiânia
22	Célia Moraes	Divorciado(a)	Natal
44	Marcelo Simas	Divorciado(a)	Natal
98	Luiza Souza	Casado(a)	Limeira

## Compressão de Dados

1- Monte a árvore de Huffman e mostre as strings compactadas de cada item abaixo:

a) AABEDEDCEBBBAAA

b) ZFZDEFZZYHH

2- Responda:

c) Como o tamanho da janela pode impactar o fator de compressão de um arquivo pelo método LZ77?

d) Como o tamanho do buffer pode impactar o fator de compressão de um arquivo pelo método LZ77?

3- Dadas as tuplas abaixo, como seria a string original sabendo que a janela tem tamanho 8 e o buffer tem tamanho 4.

(0,0,1)

(0,0,0)

(1,1,1)

(2,2,0)

(3,3,0)

(4,3,1)

(5,2,EOF)

4- Mostra as tuplas geradas pelo método LZ77 para a string abaixo, considerando uma janela de tamanho 10 e um buffer de tamanho 5.

00101011101011010001111

5- Quais as possibilidades de modificação no código do LZ78 original para que o dicionário não cresça indefinidamente?

6- Mostre o (passo-a-passo) da construção do dicionário do LZ78 para a string abaixo e também mostre a string comprimida.

00101011101011010001111