

SCC0203 - Algoritmos e Estruturas de Dados II – Profa. Cristina Oliveira

Segundo Trabalho Prático – Arquivos

Prazo máximo: 23:55h de 03/06/2012

Neste trabalho, as duas questões fornecidas são de realização obrigatória. A primeira é relativa à organização de arquivos e a segunda à compressão de dados. Ambas abordam assuntos e algoritmos vistos em sala de aula. A realização deste trabalho deve ser **em duplas**.

Para este trabalho, **não será usado o Online Judge**. Devem ser entregues três arquivos compactados no Moodle, em suas respectivas pastas: uma para cada tarefa da questão 1 e outro para a questão 2. Os arquivos compactados devem ser nomeados com o número USP dos alunos separados por “_” e conter a extensão “.zip” (ex: 1234_5678.zip) . Cada arquivo compactado deve conter:

- **Um único arquivo de código fonte “.c”**, contendo a solução do problema. O arquivo deve ser denominado da seguinte maneira: Qx_Ty_XXXX_XXXX.c, onde: x é o número da questão e y é o número da tarefa (quando aplicável). A seguir, deve-se colocar o número USP dos autores. Ainda, dentro do código fonte deve-se ter, nas primeiras linhas, um comentário contendo o nome e o número USP dos autores do trabalho.
- **O arquivo de saída** gerado pelo programa para a tarefa 1 da primeira questão. O arquivo deve ser nomeado com “tabelacampeonatofixo_XXXX_XXXX.dat”, onde XXXX deve ser substituído pelo número USP dos autores.
- **Um relatório** de, no máximo, 2 páginas, explicando a teoria aplicada à resolução do problema. Este relatório não deve conter detalhes de codificação do problema.

Observações

- Deve-se fazer a implementação em linguagem **ANSI C**.
- O sistema bloqueia entregas após o prazo estipulado. Assim sendo, não será possível entregar o trabalho após a data/horário estipulados.
- Considere as entradas e saídas **ESTRITAMENTE** nos padrões descritos. Não imprima na tela quaisquer outras coisas que não as saídas esperadas pelo programa.
- Para fins de avaliação, apenas a última submissão de cada problema é considerada. Será analisado o código fonte, especialmente os TADs utilizados. **Os trabalhos também serão processados em um verificador de plágio.**

Bom trabalho!

Questão 1

Seu Astolfo e o Campeonato Brasileiro

No dia 19 de maio, inicia-se a disputa do Campeonato Brasileiro de Futebol de 2012. Neste ano, 20 times jogam entre si, em turno e retorno, em um campeonato de pontos corridos. Ao final, aquele time que somar mais pontos, sagra-se Campeão Brasileiro de Futebol.

Seu Astolfo é um torcedor das antigas. Ele gosta de acompanhar todos os jogos do seu time pelo radinho à pilha. Para isso, Seu Astolfo usa sempre a boa e velha tabela da competição que vem no jornal da sua cidade. Este ano, entretanto, Seu Astolfo comprou seu primeiro computador. Segundo ele, seu time tem tudo para ser campeão neste ano e ele não pode perder nenhum jogo. O computador vai ser de grande valia para encontrar as datas e os horários dos jogos do seu time.

Convocamos você para ajudar o Seu Astolfo. Disponibilizamos um arquivo contendo a tabela completa do Campeonato Brasileiro de 2012 – **tabelacampeonato.dat**. Neste arquivo, cada registro corresponde a um jogo do campeonato. Cada um desses registros é composto por **8 campos**, separados por um marcador # (em ordem: **rodada, data, dia da semana, hora, mandante, visitante, estádio e cidade**). Registros estão separados pelo marcador \n. Abaixo, tem-se um exemplo de registro constante no arquivo, bem como a maneira de interpretar as suas informações.

```
3#07/jun#Qui#20:30#Botafogo#Cruzeiro#João Havelange#Rio de Janeiro#
```

1. **Rodada:** 3
2. **Data:** 07/jun
3. **Dia da semana:** quinta-feira
4. **Horário:** 20:30
5. **Mandante:** Botafogo
6. **Visitante:** Cruzeiro
7. **Estádio:** João Havelange
8. **Cidade:** Rio de Janeiro

Com este arquivo, precisamos que você faça as seguintes tarefas:

Tarefa 1) Faça um primeiro programa que converta estes arquivos para o formato de **campos e registros de tamanho fixo**. O tamanho de cada campo deve ser determinado por você. Deve ser escolhido o melhor tamanho possível para cada campo, isso é, de maneira que haja o **menor desperdício possível de espaço**. Os registros, por sua vez, terão o tamanho máximo igual à soma dos comprimentos dos 8 campos. Por exemplo, se todos os 8 campos tiverem um comprimento 20, então o comprimento do registro é de $8 * 20 = 160$. Note que os registros não devem mais estar separados por uma quebra de linha!

Seu programa deve imprimir na tela o nome de cada campo e o tamanho fixo escolhido para o mesmo, bem como o tamanho total de cada registro. Por exemplo, considerando que todos os campos sejam de tamanho 20, seu programa deve imprimir na tela:

```
Rodada 20
Data 20
Dia da semana 20
Hora 20
Mandante 20
Visitante 20
Estádio 20
Cidade 20
Registro 160
```

Ainda, seu programa deve gravar o resultado da conversão do arquivo de entrada em campos e registros de tamanho fixo em um arquivo "*tabelacampeonatofixo_XXXX_XXXX.dat*".

Tarefa 1.2) Faça um programa que receba como entrada (a partir do teclado) o nome do time para o qual o Seu Astolfo torce. Assim, lendo o arquivo "*tabelacampeonatofixo_XXXX_XXXX.dat*" que você gerou na tarefa anterior, deve-se imprimir na tela, a cada linha, os seguintes dados de todas as partidas em que o time de Seu Astolfo for **visitante**: rodada, data, horário, mandante e estádio, nessa ordem. As informações devem ser separadas por um único símbolo "\t". Por exemplo, considerando que receba-se *Cruzeiro* na entrada, a saída gerada (na tela) deverá ser:

2	26/mai	21:00	Náutico	Aflitos
3	07/jun	20:30	Botafogo	João Havelange
6	23/jun	18:30	Vasco	São Januário
8	07/jul	18:30	Internacional	Beira Rio
10	18/jul	20:30	Portuguesa	Canindé
12	25/jul	21:50	Corinthians	Pacaembu
15	08/ago	21:50	Santos	Vila Belmiro
16	11/ago	18:30	Bahia	Pituaçu
18	19/ago	16:00	Coritiba	Couto Pereira
20	29/ago	18:30	Atlético-GO	Serra Dourada
23	09/set	18:30	Sport	Ilha do Retiro
24	12/set	18:30	Figueirense	Orlando Scarpelli
26	23/set	18:30	São Paulo	Morumbi
28	06/out	16:00	Grêmio	Olímpico
30	17/out	18:30	Flamengo	João Havelange
32	24/out	21:00	Palmeiras	Pacaembu
33	27/out	16:00	Ponte Preta	Moisés Lucarelli
36	18/nov	19:30	Fluminense	João Havelange
38	02/dez	16:00	Atlético-MG	Arena do Jacaré

Para essa tarefa, deve-se considerar que possui-se uma **memória principal limitada**, que suporta apenas **3 registros** armazenados. Deve-se, ainda, fazer o menor número possível de acessos ao arquivo. Seu programa deve imprimir na tela, ao final, logo abaixo da lista de informações solicitadas, o número total de consultas em arquivo realizadas para a leitura das informações solicitadas, tal como no exemplo abaixo.

Acessos 12345

Questão 2

Comprimindo os dados do Seu Astolfo

Seu Astolfo, agora, enfrenta um novo problema. De tanto baixar da internet textos, fotos e vídeos sobre a história do seu time, o disco rígido do seu computador recém adquirido está muito próximo de seu limite máximo.

Procurando na internet por uma solução para isso, Seu Astolfo ouviu falar de “compressão de arquivos”. Ele, muito interessado, procurou mais sobre o assunto e ouviu que um senhor chamado David Huffman havia proposto uma solução muito simples para a codificação de arquivos que, ao final, reduzia o seu tamanho.

Seu Astolfo, então, pediu que você o ajudasse com esse problema. Usando a **Codificação de Huffman**, você deve fazer um programa que leia um arquivo, cujo caminho seja passado como entrada (teclado), compacte-o e seja capaz de descompactá-lo.

O arquivo deverá ser lido byte a byte, isto é, deve-se considerar a existência de, no máximo, 256 símbolos. Seu programa deverá realizar uma primeira leitura do arquivo para contar o número de ocorrências de cada símbolo e, a partir desta informação, construir uma codificação de Huffman para o conjunto de dados. Uma segunda passada pelo arquivo será necessária para comprimi-lo.

Para permitir a decodificação do arquivo, será necessário armazenar uma tabela contendo o número de ocorrências de cada símbolo no início do arquivo comprimido (antes dos dados codificados). Dessa forma, o decodificador poderá recuperar do arquivo a frequência de cada um dos símbolos e, conseqüentemente, reconstruir a mesma codificação de Huffman utilizada na compressão. Uma vez restaurada a codificação utilizada, a descompressão das informações pode ser realizada facilmente.

O seu programa deverá ler o arquivo de entrada, cujo caminho foi especificado na entrada, e comprimi-lo, salvando-o com o nome “*arquivocomprimido.huf*”. Em seguida, seu programa deve ler o arquivo “*arquivocomprimido.huf*” e descomprimi-lo, salvando-o com o nome “*arquivodescomprimido.huf*”.