

Universidade Federal da Bahia
Graduação em Ciência da Computação
MATA40 - Estruturas de Dados e Algoritmos I
Primeiro Trabalho Prático
Prof. Flávio M. Assis Silva
Semestre 2011.1 - 9 de maio de 2011

Dicionário

1 Descrição Geral do Trabalho

Neste trabalho o aluno implementará um programa que lerá dados de um dicionário e permitirá que operações sobre este dicionário sejam feitas. O programa terá como entrada um conjunto de palavras em um idioma, chamado de *idioma origem*, suas traduções em outro idioma, chamado de *idioma destino*, e um conjunto de comandos a serem executados. Os comandos são usados para se formularem as operações sobre o dicionário. O programa deve ler da entrada padrão e escrever os resultados na saída padrão.

Cada definição de palavra conterá os seguintes dados: palavra em um idioma (inglês, por exemplo), classe da palavra (substantivo, adjetivo ou verbo) e um conjunto de traduções para a palavra. Os tipos de operações que poderão ser realizadas sobre o dicionário são, dentre outras: listar traduções de uma palavra; e listar palavras do idioma origem em ordem alfabética. A seção 2 detalha os tipos de operações que o programa deve suportar.

O programa deverá manter as seguintes estruturas de dados:

1. um vetor de apontadores, de tamanho *NTABHASH* (configurável). Cada entrada do vetor apontará para uma lista que conterá palavras do idioma origem. Uma palavra estará na lista correspondente à entrada i se e somente se o resultado da aplicação de uma determinada função de *hashing* sobre esta palavra tiver como resultado o valor i . Seja $p = p_1p_2...p_m$ uma palavra p formada por m caracteres p_1 a p_m . Cada p_i pode ser ou uma letra ou um dígito. A função de *hashing* a ser adotada será:

$$hash(p) = (\sum_{i=1}^m ordc(p_i)) \bmod NTABHASH ,$$

onde $ordc(p_i)$ é definida da seguinte forma: se p_i é uma letra, então $ordc(p_i)$ corresponde à posição da letra p_i no alfabeto (considere que a posição da letra 'a' é 1); se p_i for um dígito, $ord(p_i)$ será igual ao valor de p_i .

Os índices deste vetor vão de 0 a $NTABHASH - 1$. $NTABHASH$ deve ter valor inicial 11. O programa, no entanto, deve ser feito de forma que este valor possa ser alterado.

2. cada nó das listas indicadas no item acima deve conter, além da palavra (e de apontadores para as listas descritas no item (1) e nos itens abaixo): um apontador para uma lista de traduções e um campo indicando a classe da palavra (se substantivo, adjetivo ou verbo). A lista de traduções será uma lista encadeada simples. Cada nó da lista conterá uma tradução (cadeia de caracteres) para a palavra no idioma original.
3. o programa deve manter todas as palavras do idioma origem que são de uma mesma classe unidas em uma lista própria. Haverá, com isto, três novas listas, uma unindo os substantivos, outra unindo os adjetivos e outra unindo os verbos. Estas listas devem ser ordenadas em ordem alfabética crescente, devem ser duplamente encadeadas e devem usar os mesmos nós da lista do item (1) acima (ou seja, as palavras do idioma origem devem ser armazenadas apenas uma vez no programa).
4. o programa deve manter também todas as palavras do idioma origem em uma lista duplamente encadeada adicional, que unirá as palavras em ordem alfabética crescente.

Por exemplo, suponha que inglês seja o idioma origem, português seja o idioma destino e as seguintes traduções tenham sido dadas: *veículo de passeio* e *carro* para a palavra *car*. Considerando que $NTABHASH$ tenha valor 11, o valor da função de *hashing* aplicada à palavra *car* será 0. O vetor do item (1) terá, em sua entrada de índice 0, um apontador para uma lista contendo um único nó. Este nó conterá os seguintes dados: a palavra *car*, um campo indicando que se trata de um substantivo e um apontador para uma lista de traduções. Esta lista conterá dois nós, um contendo a tradução *veículo de passeio* e outro contendo a tradução *carro*. Adicionalmente, o nó que armazena a palavra *car* será um nó de uma lista de substantivos e de outra lista que conterá as palavras em ordem alfabética. Estas listas são duplamente encadeadas e devem utilizar o próprio nó em que a palavra *car* está armazenado. Ou seja, não devem ser criadas listas adicionais separadas. Os apontadores para manutenção destas listas estarão implementados em um único nó.

2 Formato da Entrada

A entrada constará de uma seção de traduções seguida por uma seção de operações. A seção de traduções conterá uma linha contendo um valor inteiro, seguida por um conjunto de traduções. O valor inteiro na primeira linha conterá o número de traduções que haverá em seguida. Cada uma das traduções conterá: uma linha contendo uma palavra no idioma origem, seguida por um espaço, seguido por uma das letras 's', 'a' ou 'v', seguida por um espaço, seguido por

um número inteiro. As letras 's', 'a' e 'v' indicam, respectivamente, que a palavra se trata de um substantivo, adjetivo ou verbo. O número inteiro após a letra indica o número de traduções para a palavra. As traduções aparecerão nas linhas seguintes, uma por linha. Considere que o tamanho máximo de uma palavra no idioma origem é de 30 caracteres. Cada tradução pode ser uma palavra ou um conjunto de palavras, mas cada tradução terá, no máximo, 50 caracteres (incluindo espaços).

Na sequência de comandos, os seguintes comandos e suas estruturas poderão ocorrer:

1. **lista palavras no idioma origem:** este comando consistirá de uma linha contendo a letra 'l', seguida de um espaço, seguido ou da letra 'c' ou 'd'. Este comando lista as palavras do idioma origem. Se a segunda letra for 'c', as palavras devem ser mostradas na ordem crescente. Se for 'd', as palavras devem ser mostradas na ordem decrescente.
2. **lista palavras em ordem alfabética:** este comando consistirá de uma linha formada pela letra 'o', seguida de um espaço, seguido da letra 'c' ou 'd', seguida de um espaço, seguido de uma letra, seguida de um espaço, seguido de outra letra. Seja *letra3* a terceira letra do comando e seja *letra4* a quarta.

Se a segunda letra for 'c', o comando listará as palavras do dicionário, em ordem crescente, que comecem com uma das letras entre *letra3* e *letra4*, incluindo *letra3* e *letra4*. Se a segunda letra for 'd', o comando listará as palavras do dicionário, em ordem decrescente, que comecem com uma das letras entre *letra3* e *letra4*, incluindo *letra3* e *letra4*. *letra3* será igual a *letra4* ou virá antes de *letra4* no alfabeto.
3. **lista traduções:** este comando consistirá de uma linha contendo a letra 't', seguida de um espaço, seguido de uma palavra do idioma origem. O comando deve apresentar as traduções para aquela palavra.
4. **lista palavras por classe:** este comando consistirá de uma linha contendo a letra 'a', seguida de um espaço, seguido da letra 's', 'a' ou 'v', seguida de um espaço, seguido da letra 'c' ou 'd', seguida de um espaço, seguido de uma letra, seguida de um espaço, seguido de outra letra. O comando deve retornar palavras do idioma origem que são substantivos, adjetivos ou verbos, a depender de a segunda letra ser, respectivamente, 's', 'a' ou 'v'. As palavras retornadas devem estar em ordem crescente, se a terceira letra for 'c', ou decrescente, se esta letra for 'd'. Sejam *letra4* e *letra5*, respectivamente, a quarta e a quinta letra da operação. As palavras retornadas serão aquelas que começam com uma letra que estiver, na ordem alfabética, entre *letra4* e *letra5* (incluindo estas letras). *letra4* será igual a *letra5* ou virá antes de *letra5* no alfabeto.
5. **lista palavras por índice do vetor:** este comando consistirá de uma linha contendo a letra 'i', seguida de um espaço, seguido de um número

inteiro entre 0 e $TAMTABHASH - 1$. Este comando retorna as palavras que estão armazenadas na lista correspondente à entrada cujo índice é o valor inteiro que aparece no comando. As palavras devem ser retornadas em ordem alfabética crescente.

6. **consulta classe de palavra:** este comando consistirá de uma linha contendo a letra 'c', seguida de um espaço, seguido de uma palavra. Este comando retornará a classe (substantivo, adjetivo ou verbo) da palavra digitada no comando.
7. **remove palavra:** este comando consiste de uma linha contendo a letra 'r', seguida de um espaço, seguido de uma palavra. Este comando remove a palavra digitada do dicionário.
8. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

A entrada somente conterá caracteres minúsculos.

3 Formato da Saída

Cada comando deverá apresentar a saída da seguinte forma:

1. **lista palavras no idioma origem:** este comando deve apresentar uma palavra por linha, sem espaço após cada palavra. Caso o dicionário esteja vazio, o comando não gera saída (nem espaço em branco).
2. **lista palavras em ordem alfabética:** este comando deve apresentar uma palavra por linha, sem espaço após cada palavra. Caso o dicionário esteja vazio, o comando não gera saída (nem espaço em branco).
3. **lista traduções:** este comando deve apresentar uma tradução por linha (exatamente como aparece na entrada, com exceção de que todas as palavras devem aparecer em letras minúsculas). Caso o dicionário não contenha palavra digitada no comando, nenhuma saída deverá ser gerada (nem espaço em branco).
4. **lista palavras por classe:** este comando deve apresentar uma palavra por linha, sem espaço após cada palavra. Caso não existam palavras na classe desejada, o comando não gera saída (nem espaço em branco).
5. **lista palavras por índice do vetor:** este comando deve apresentar uma palavra por linha, sem espaço após cada palavra. Caso o vetor não contenha palavras associada ao índice digitado no comando, nenhuma saída será gerada (nem espaço em branco).
6. **consulta classe de palavra:** se a palavra desejada estiver no dicionário, este comando retorna a sua classe. A classe deve aparecer da seguinte maneira: *substantivo*, *adjetivo* ou *verbo*. Caso o dicionário não contenha a palavra desejada, o comando não gera saída (nem espaço em branco).

7. **remove palavra:** este comando não gera saída.

8. **término da sequência de comandos:** este comando não gera saída.

Importante: o programa não deve gerar nenhum caractere a mais na saída, além dos indicados acima.

Não deve haver espaço entre linhas na saída. A saída deve apresentar os caracteres em letras minúsculas.

4 Observações

Trabalho individual.

Data de entrega: 08/06/2011

Linguagens de programação permitidas: Pascal, C, C++, Java

Observação Importante: Somente trabalhos feitos utilizando-se os seguintes compiladores serão aceitos:

- Pascal: FreePascal
- C: gcc ou djgpp
- C++: g++ ou djgpp
- Java: compilador java do JDK (mais recente)

Não serão compilados trabalhos em outros compiladores! Erros ocasionados por uso de diferentes compiladores serão considerados erros do trabalho!

O aluno deverá armazenar seu trabalho, protegido por senha, na página da disciplina e enviar a senha para o email do professor (o aluno deve utilizar *zip*).