Universidade Federal da Bahia
Graduação em Ciência da Computação
MATA54 - Estruturas de Dados e Algoritmos II
Segundo Trabalho Prático
Prof. Flávio Assis
Semestre 2017.1 - 2 de agosto de 2017

Auxílio à Redação

1 Descrição Geral do Trabalho

Neste trabalho o aluno implementará um programa para corrigir palavras e auxiliar a composição de frases. A funcionalidade a ser implementada é similar à que é tipicamente encontrada associada a teclados em celulares. À medida em que palavras são digitadas, o programa corrige palavras escritas incorretamente, aprende novas palavras e indica possíveis próximas palavras na composição de frases.

A correção de palavras será feita usando uma *base* de palavras. As palavras contidas nesta base serão as consideradas corretas pelo programa. Durante a execução do programa novas palavras serão inseridas na base.

Especificamente, o usuário irá digitar uma sequência de palavras. Para cada palavra digitada, o programa tomará uma ação de acordo com o descrito abaixo:

1. se a palavra for considerada incorreta (não estiver na base), o programa deve propor três palavras que são consideradas alternativas corretas para a palavra;

2. se a palavra considerada incorreta pelo programa for considerada correta pelo usuário, o programa irá inserir a palavra na base;

3. se a palavra for correta, o programa irá sugerir até três palavras que são candidatas a próximas palavras a serem digitadas pelo usuário.

2 Procedimentos Gerais

2.1 Alternativas a Palavras Consideradas Incorretas

Uma palavra p da base será considerada alternativa para uma palavra q, considerada incorreta, se: p tiver o mesmo número de caracteres de q; e p e q diferirem entre si de no máximo um caractere. Dentre o conjunto de palavras que podem ser alternativas na base, o programa irá propor as três mais frequentemente digitadas pelo usuário. O programa deve, portanto, manter um controle sobre as frequências com que cada palavra é digitada pelo usuário.

Caso não haja três palavras que atendam os critérios acima, o programa irá sugerir as palavras que atendem os critérios e quaisquer outras palavras da base para completar as três sugestões. Se não houver palavras suficientes para completar as três sugestões, o programa deve apresentar as palavras existentes na base.

2.2 Sugestão de Próximas Palavras a Serem Digitadas

No caso de o usuário digitar uma palavra correta, o programa deverá propor as três palavras que mais frequentemente foram digitadas pelo usuário após a palavra correta.

Caso não haja três palavras que atendam o critério acima, o programa irá sugerir quaisquer outras palavras da base para completar as três sugestões. Caso não haja três palavras, o programa apresenta as palavras existentes na base.

2.3 Persistência

O estado dos dados do programa deve ser mantido persistente. Ou seja, inicialmente o programa deve criar os arquivos necessários para manter os dados persistentes. A partir de então, quaisquer alterações nos dados devem ser refletidas no conteúdo dos arquivos. O estado dos dados deve ser mantido de uma invocação a outra do programa. Quando uma nova sequência de testes do programa for ser criada, os arquivos com os dados serão apagados.

Todos os arquivos utilizados pelo programa para armazenar os dados deverão ter extensão $".dat"$.

2.4 Considerações Gerais Adicionais

Apesar de ter que manter os dados persistentes, pode-se considerar que os dados utilizados pelo programa cabem na memória principal.

As estruturas de dados e algoritmos utilizados pelo aluno são de livre escolha. O aluno deve, no entanto, elaborar sua implementação utilizando da melhor forma possível o conteúdo da disciplina.

3 Formato da Entrada e Saída

O programa deve ler da entrada padrão e escrever na saída padrão. Somente letras ocorrerão em palavras, sem acento e sem cedilha. Todos os caracteres na entrada serão minúsculos. Uma palavra poderá conter, no máximo, 30 caracteres. Todas as saídas geradas devem apenas conter letras minúsculas. Em cada instante, o programa deve sempre considerar a situação de a base estar vazia.

A entrada consistirá de uma sequência de operações. O conjunto de operações a serem implementadas e seus formatos são os seguintes:

1. **inserção de palavras na base:** esta operação conterá inicialmente uma linha contendo a letra i, seguida de outra linha contendo um número

inteiro maior que zero. Esse número indicará o número de palavras que serão digitadas a seguir, uma por linha.

Cada palavra digitada será uma palavra considerada correta a ser inserida na base. O programa somente deve inserir na base palavras que não se encontrem já na base.

Esta operação *não será* considerada para determinação da frequência de digitação das palavras.

2. **digitação de palavra:** esta operação terá duas linhas: uma contendo a letra d e outra contendo uma palavra.

   Se a palavra estiver incorreta, o programa deve imprimir a sequência de caracteres *"Palavra desconhecida. Possiveis correcoes:"*, seguida de um espaço, seguido de (até) três palavras que são alternativas para correção da palavra (como descrito acima). Estas palavras devem ser apresentadas na mesma linha, separadas por um espaço. Em seguida, o usuário deverá digitar ou a palavra que ele digitou inicialmente ou uma das palavras propostas pelo programa. No primeiro caso, o sistema deve considerar a palavra digitada pelo usuário como uma nova palavra correta e deve inseri-la na base. No segundo caso, o programa deve imprimir a sequência de caracteres *"Proximas palavras:"*, seguida de um espaço, seguido de três palavras, que são sugestões de próxima palavra a ser digitada pelo usuário (como descrito acima). Essas três palavras devem ser apresentadas na mesma linha, separadas por um espaço.

   No momento em que uma palavra for considerada correta, seja por ter sido inicialmente digitada corretamente ou por ter sido uma correção aceita pelo usuário, a palavra passará a ser considerada digitada, para efeito de cálculo de frequência de uso de palavras e de palavras em sequência. Uma palavra correta será considerada digitada após a última palavra correta digitada.

   As palavras apresentadas devem aparecer na ordem decrescente de frequência (de uso ou de correspondência com a palavra anterior). Palavras impressas apenas para completar as três sugestões podem aparecer em qualquer ordem, ao final na linha.

3. **impressão da base de palavras com frequências:** esta operação conterá apenas uma linha, contendo a letra f. Esta operação listará as palavras em ordem alfabética, uma por linha. Cada linha terá a palavra, seguida de um espaço, seguido do número de vezes em que a palavra foi digitada.

4. **frequência de próximas palavras:** esta operação conterá uma linha com a letra p, seguida de outra linha contendo uma palavra. Esta operação apresentará as palavras mais frequentes utilizadas após a palavra indicada, até no máximo três. Cada linha da saída conterá uma palavra, seguida de um espaço, seguido do número de vezes em que a palavra foi utilizada após a palavra apresentada como parâmetro para a operação.

5. **término da sequência de comandos:** a sequência de comandos será terminada por uma linha com a letra 'e'.

4 Observações

Trabalho individual. O aluno deve entregar seu trabalho através da plataforma Moodle, em um único arquivo. Este arquivo deve conter:

1. os arquivos com a implementação (apenas código fonte);
2. um texto, em formato pdf, descrevendo, de forma geral, as estruturas de dados e os algoritmos utilizados.

Data de entrega: 30/08/2017

Linguagens de programação permitidas: C, C++, Java ou Python.

Observação Importante: Para as linguagens C, C++ e Java, somente trabalhos feitos utilizando-se os seguintes compiladores serão aceitos:

- 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!