



Disciplina Algoritmos e Estruturas de Dados II	Turma TM3-4
Professores: William Robson Schwartz Thiago Noronha	

Data de entrega: 01/07/2018 até às 23:55

## Trabalho Prático II

### 1 Teclado com Autocomplete

A finalidade nesse trabalho é desenvolver um sistema de auto completção de teclado. Além disso, é esperado que o aluno desenvolva a prática de modelagem em tipos abstratos de dados (TADs) e a prática de programação, em particular para algoritmos de busca e ordenação.

### 2 Descrição

Dado um dicionário de palavras, o sistema deverá criar uma tabela hash para indexar todas as palavras do dicionário de acordo com seus prefixos. Após a indexação de todas as palavras, o sistema deverá atender a requisições do usuário. Quando o usuário digitar uma letra, o programa deverá procurar na tabela hash por palavras que contenham aquela letra inicial e exibi-las na tela sem a necessidade de apertar a tecla *Enter*. Quando o usuário digitar a segunda letra, o programa deverá pesquisar na tabela por palavras que iniciem com as duas letras passadas. A tabela hash deverá ser criada para até  $n$  letras iniciais. O programa deverá mostrar todas as sugestões de palavras que contenham o prefixo passado em **ordem alfabética**. Caso não existam sugestões, o sistema deverá informar isso ao usuário através de uma mensagem.

### 3 Entrada

O seu programa receberá um arquivo contendo um conjunto de palavras do dicionário. Além disso, um inteiro  $n$  será passado indicando quantas letras deverão ser utilizadas para fazer a indexação na tabela. Por exemplo, caso 'n' seja 3, então todos os prefixos de tamanho até 3 (gab, ren, joa, wil, thi, pro) deverão ser indexados.

Caso o usuário digite mais letras do que o tamanho 'n' passado, você não deverá mostrar nenhuma sugestão. Lembre-se de que as sugestões deverão ser mostradas em tempo real sem a necessidade da tecla enter ( **Dica: você pode usar o comando `getchar()`**). Além disso, a tecla *Backspace* deverá funcionar normalmente.

O programa não deverá fazer distinção de letras maiúsculas ou minúsculas e deverá encerrar quando o usuário apertar a tecla *Enter*. Arquivos de dicionário serão disponibilizados para que você teste o seu código.

Exemplo de execuções do programa:

```
>>> Digite uma palavra: est
Sugestões: estante, estatística, estudo
Programa finalizado!!

>>> Digite uma palavra: pr
Sugestões: pragmático, premissa, problema, programação
Programa finalizado!!
```

## 4 Perguntas

Na sua documentação, você deverá incluir no texto a resposta para as seguintes perguntas:

1. Qual o custo de montar a tabela hash?
2. Qual o custo de buscar uma palavra?
3. Qual o custo para ordenar as sugestões?
4. Qual o custo de inserir uma nova palavra?
5. Qual o custo de remover uma palavra?
6. Qual o custo total do seu código?
7. Qual função hash você escolheu?
8. Como seu sistema se comporta no caso de haver colisões?

## 5 Submissão

A forma de submissão do trabalho será via Moodle. Ao receber seu código, nós compilaremos seu programa em um ambiente Linux com o seguinte comando:

```
gcc *.c -o tpAEDSII
```

Em seguida, executaremos seu programa com o seguinte comando:

```
./tpAEDSII arquivoDeEntrada n
```

Ou seja, utilizem a linha de comando. Não há problema algum em desenvolver o trabalho no Windows, apenas tome cuidado para não utilizar bibliotecas que não compilam no Linux (por exemplo, <windows.h>)

## 6 Comentários Gerais

1. Você deverá entregar uma documentação explicando as estratégias utilizadas para a realização do trabalho.
2. Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está longe.
3. Clareza, indentação e comentários no programa também serão avaliados.
4. **SOFTWARE DE DETECÇÃO DE PLÁGIOS SERÃO UTILIZADOS, PORTANTO NÃO COPIEM O TRABALHO DO COLEGA.**
5. Não se desespere, faça uma parte de cada vez. Só continue o trabalho quando tiver certeza que o que fez está funcionando corretamente, teste bem cada função que fizer.
6. Também avaliaremos a adequação ao padrão de entrega.
7. O trabalho pode ser feito de forma individual.