

# Trabalho 1 - Dicionário de Anagramas

Prof. Dr. Juliano Henrique Foleis

## Anagramas

Um anagrama de uma palavra  $p$  é definido como uma outra  $q$  palavra do dicionário  $D$  que satisfaz a seguinte restrição:

$$(q = \text{Perm}(p)) \wedge (p \in D) \wedge (q \in D)$$

tal que  $\text{Perm}(p)$  é alguma permutação da palavra  $p$  e  $D$  é o dicionário.

## Exemplos

Os anagramas da palavra *cebola* são *cebola* e *cabelo*.

Os anagramas da palavra *amor* são *amor*, *armo*, *oram*, *ramo*, *roam*, e *roma*. (sim *roam* é uma palavra da língua portuguesa.)

## Descrição do Trabalho

Você deve implementar um dicionário de anagramas para palavras da língua portuguesa em C. Seu programa deve receber uma única palavra pela linha de comando (sem acentos) e imprimir na saída a lista de todos os seus anagramas em ordem alfabética, incluindo a própria palavra de entrada. Cada anagrama deve ser impresso em sua própria linha. Desconsidere a diferença entre maiúsculas e minúsculas. Por exemplo, caso o usuário entre com *CEBOLA*, a saída deve ser *cebola* e *cabelo*.

A busca pelos anagramas deve ser realizada em  $\Theta(\lg n)$ , tal que  $n$  é o número de palavras no dicionário da língua portuguesa. O pré-processamento deve ocorrer no máximo em tempo  $\Theta(n \lg(n))$ .

O dicionário de palavras é fornecido. Para evitar ter que lidar com UTF-8 em C, o dicionário não contém acentos.

## Dicas

- Quando ordenadas por letras, palavras anagramas são representadas pela mesma string.
- A busca ser realizada em  $\Theta(\lg n)$  quer dizer que você não deve usar uma busca sequencial para pesquisar os anagramas!
- O dicionário fornecido (br.txt) tem todas as palavras da língua portuguesa. Cada palavra está em uma linha terminada por `\n`. Ele está codificado em ASCII.

## Avaliação

- O trabalho pode ser feito individualmente ou em duplas.
- O trabalho deve ser implementado exclusivamente em linguagem C.
- A organização do código em módulos e TADs será avaliada.
- Somente as seguintes bibliotecas podem ser usadas: *stdlib*, *stdio* e *string*. **As funções *bsearch* e *qsort* da *stdlib* não podem ser usadas.**

- A equipe deve implementar todas estruturas na “mão”.
- A entrega do trabalho deverá ser feita impreterivelmente até dia 26/04/2021. Trabalhos entregues após esta data serão desconsiderados.
- Em caso de plágio o trabalho será anulado. Lembre-se, copiar de outro grupo é plágio, tanto quanto copiar códigos prontos da internet. No caso de cópia de outro(s) grupo(s), a nota de todos os grupos será anulada! No caso de cópia de códigos prontos da internet sua nota será anulada. Em caso de plágio não há recuperação de nota.
- O trabalho deve ser entregue em formato .tar.gz ou .zip contendo o código-fonte, um Makefile e o PDF de um relatório de implementação, como especificado abaixo.
- O relatório deve ter no máximo 10 páginas, descrevendo a implementação desenvolvida. Esta descrição deve descrever o problema que foi resolvido, bem como indicar como que o código foi modularizado e quais foram os TADs desenvolvidos. Inclua como foi realizada a divisão das tarefas entre os membros da equipe. Inclua também alguns exemplos de entrada e saída do seu programa. Não esqueça de indicar no relatório o nome dos alunos da equipe.
- Os trabalhos devem ser entregues via Moodle. Somente um aluno do grupo precisa enviar o trabalho.

Bom Trabalho!