

# Aula Prática 10 - Estruturas de Dados I (BCC202)

Marco Antonio M. Carvalho  
Universidade Federal de Ouro Preto  
Departamento de Computação

27 de outubro de 2021

## Instruções

- Siga atentamente quanto ao formato da entrada e saída de seu programa, exemplificados no enunciado;
- Durante a correção, os programas serão submetidos a vários casos de testes, com características variadas;
- A avaliação considerará o tempo de execução e o percentual de respostas corretas;
- Eventualmente realizadas entrevistas sobre os estudos dirigidos para complementar a avaliação;
- Considere que os dados serão fornecidos pela entrada padrão. Não utilize abertura de arquivos pelo seu programa. Se necessário, utilize o redirecionamento de entrada;
- Os códigos fonte serão submetidos a uma ferramenta de detecção de plágios em software;
- Códigos cuja autoria não seja do aluno, com alto nível de similaridade em relação a outros trabalhos, ou que não puder ser explicado, acarretará na perda da nota e frequência;
- Códigos ou funções prontas específicos de algoritmos para solução dos problemas elencados não são aceitos;
- Não serão considerados algoritmos parcialmente implementados.

# 1 Árvores Binárias

Você foi convidado a escrever um programa que liste todas as diferentes palavras que existem em um texto, em ordem alfabética. Neste caso, uma palavra é definida como uma sequência de letras, maiúsculas ou minúsculas. Palavras com apenas uma letra também deverão ser consideradas. Portanto, seu programa deverá ser "CaSe InSeNsItIvE". Por exemplo, palavras como "Apple", "apple" ou "APPLE" deverão ser consideradas como a mesma palavra.

## Especificação da Entrada

A entrada contém no máximo 10000 linhas de texto, cada uma delas com no máximo 200 caracteres. O fim de entrada é determinado por EOF.

## Especificação da Saída

Você deve imprimir uma lista de diferentes palavras que aparecem no texto, uma palavra por linha. Todas as palavras devem ser impressas com letras minúsculas, em ordem alfabética. Haverá no máximo 5000 palavras distintas.

## Exemplo de Entrada

```
Ex(*$a#.mpl.e:
```

```
Adventures in Disneyland
```

```
Two people were going to Disneyland when they came to a fork in the road.
```

```
The sign read: "Disneyland LEFT."
```

```
So they went home.
```

## Exemplo de Saída

```
a
adventures
came
disneyland
e
ex
fork
going
home
in
left
mpl
people
read
road
sign
so
```

the  
they  
to  
two  
went  
were  
when

### Estrutura do código

O código-fonte deve ser modularizado corretamente conforme os arquivos de protótipo fornecidos. Uma árvore binária deve ser criada, preenchida e percorrida para determinação da solução. Nenhum algoritmo de ordenação deve ser utilizado. Os procedimentos *TNo\_Cria* e *TArvore\_Inicia*, bem como o TAD *TArvore* são similares aos vistos na aula teórica, porém, precisam ser adaptados para lidarem com *strings*. Adicionalmente, uma das diferentes funções de inserção deve ser escolhida para solução do problema, bem como uma função de caminhamento na árvore.

### Diretivas de Compilação

```
$ gcc arvore.c -c  
$ gcc principal.c -c  
$ gcc arvore.o principal.o -o programa
```