

Modalidade: em grupo (no máximo 3 alunos por grupo)

Data de entrega: 26/junho até 23:59 na tarefa do Microsoft Teams. Entregar UM ÚNICO ARQUIVO COMPACTADO .ZIP CONTENDO APENAS OS ARQUIVOS FONTES e o arquivo *readme*.

Fazer um programa em linguagem C para gerenciar um dicionário, implementado como uma árvore PATRICIA, conforme a 1ª **abordagem** apresentada nos *slides* vistos em aula.

O programa deve seguir as seguintes regras:

- O programa deve ler um arquivo .txt contendo palavras e construir a árvore PATRICIA correspondente.
- As palavras no arquivo texto não estão necessariamente ordenadas alfabeticamente
- O programa não deve fazer distinção entre maiúsculas e minúsculas
- Palavras com acento e caracteres especiais devem ser ignoradas (não inseridas na árvore PATRICIA)
- Nos arquivos textos de entrada há uma única palavra por linha.

O programa deve permitir ao usuário as seguintes operações:

- Carregar arquivo texto: lê um arquivo contendo palavras (1 por linha) e constrói a árvore PATRICIA
- Consultar palavra: dada a palavra *w*, imprimir na tela as palavras do dicionário que tem *w* como prefixo separadas por vírgula, em ordem alfabética, limitadas ao número de 10 palavras. Por exemplo: considerando a árvore da Figura 1, e sendo *w* = **neuro**, a saída é: **neurociencia, neurologia, neurologista, neuronio, neuropatia, neuropsicologia**
- Imprimir dicionário: imprime todas as palavras armazenadas no dicionário em ordem alfabética. Por exemplo, considerando a árvore da Figura 1, a saída é:
angstron, angular, borda, brahma, brahms, brasil, neurociencia, neurologia, neurologista, neuronio, neuropatia, neuropsicologia
- Carregar arquivo de *stopwords*: remove do dicionário todas as palavras contidas no arquivo carregado (1 palavra por linha)
- imprimir subárvore por níveis: dada um prefixo *w*, imprimir todas as subárvores abaixo de *w* por nível, mostrando o número filhos em cada subárvore. Por exemplo, considerando a árvore PATRICIA mostrada na Figura 1 e
w = **neuro**, saída é:
nível 0: (-:3)
nível 1: (neuro:4)
nível 2: (ciencia:0) (logi:2) (nio:0) (p:2)
nível 3: (a:0) (sta:0) (atia:0) (sicologia:0)

Considerando *w* = **bra**, saída é:

```
nível 0: (-:3)
nível 1: (b:2)
nível 2: (ra:2)
nível 3: (hm:2) (sil:0)
nível 4: (a:0) (s:0)
```

O arquivo *readme* deve conter:

- nomes dos componentes do grupo
- opções implementadas, seguidas de uma breve explicação para o usuário do programa

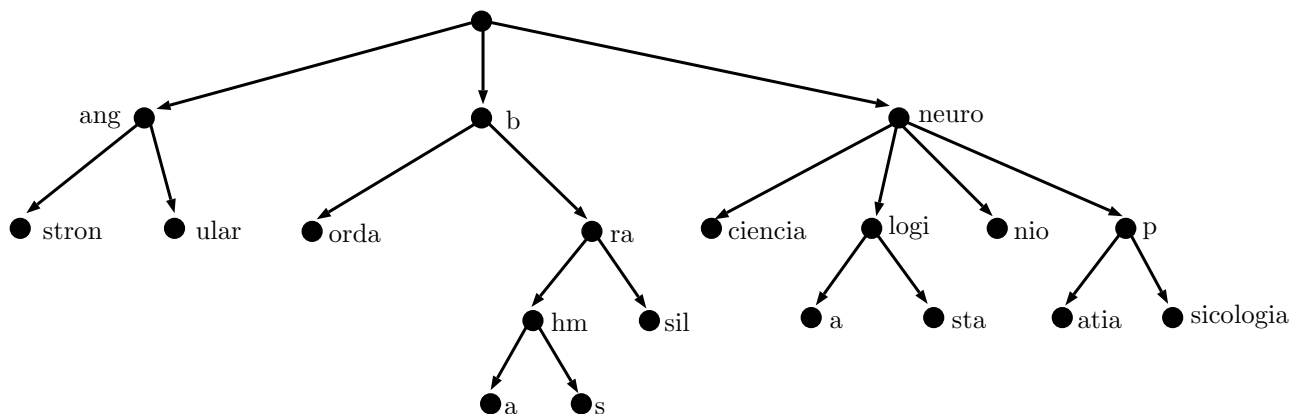


Figura 1: Exemplo de árvore PATRICIA

Critérios de avaliação:

- o trabalho deverá ser apresentado oralmente em grupo, seguido de arguição sobre o código/teoria relacionada.
 - a nota referente à apresentação oral é individual conforme desempenho do aluno na arguição.
 - o aluno do grupo que não esteja presente durante a apresentação oral, não terá a nota correspondente.
- documentação/organização do código-fonte: peso 1
- arquivo *readme*: peso 1
- implementação correta das funcionalidades: 8

Em todas as funcionalidades, a avaliação será feita considerando corretude e qualidade do código. Trabalhos copiados ou plagiados receberão nota ZERO!