UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

SSC0603 – Estrutura de Dados I – 2024-2 (ED1)

Professor responsável: *Fernando Santos Osório*
Semestre: 2024/2
Horário: Quinta 13h/14h (Prova P2)

Wiki: SSC-603-2024(FOsorio)
Web: ***http://www.icmc.usp.br/~fosorio /***
DATA PROVA: 28/11/2024 - PROVA P2-FINAL

NRO. USP: **< Colocar o seu NUSP no programa fonte>** COMO COMENTÁRIO no CÓDIGO

NOME : **< Colocar o seu Nome no programa fonte>** COMO COMENTÁRIO no CÓDIGO

>> COLOCAR SEU NOME E NRO. USP COMO COMENTÁRIO DO PROGRAMA FONTE ENTREGUE!

PROVA PROVA P2 FINAL – SSC0603 – ED1

Q1 (Questão única). Implemente o seguinte programa descrito abaixo de acordo com o especificado. O programa deve ser enviado para o RunCodes da Disciplina SSC0603 (Prova P2 FINAL)
RunCodes SSC0603 - https://runcodes.icmc.usp.br/offerings/view/62 - Regras da Prova no final deste texto.

O programa da prova é a implementação de um dicionário de tradução US=>PT e PT=>US baseado no USO DE ÁRVORES BINÁRIAS ORDENADAS (TAD visto em aula ou usando outro TAD qualquer desde que tenha alocação dinâmica de nodos e a árvore seja binária e ordenada). A árvore não precisa ser balanceada, é uma ABO simples.

Descrição: Criar um programa "modular" (com sub-rotinas, conforme descrição mais abaixo), que realize as seguintes tarefas:

(1) Ler o arquivos texto de entrada que possui o dicionário de palavras. O arquivo de dicionário será denominado de "dict.txt". O formato do arquivo é o seguinte: sequência de linhas com 2 palavras separadas por vírgula, seguidas de mais uma vírgula e de um '#' (Tipo arquivo CSV com 3 colunas: 1ª. coluna é a palavra em inglês, a 2ª. coluna é a palavra em português, e a 3ª. coluna tem um '#' que marca o fim de linha). Formato das linhas de tradução de palavras/termos:

<palavra_ou_termo_em_inglês>,<palavra_ou_termo_em_português>,#

Veja um exemplo do arquivo "dict.txt" abaixo:

```
the,o+a,#
book,livro,#
is,esta,#
on,sobre,#
table,mesa,#
#
```

Você terá que "extrair da linha" a primeira palavra (US – Inglês) e a segunda palavra (PT – Port.) Pode ler o arquivo usando fscanf ou fgets, como achar melhor.

INFORMAÇÕES IMPORTANTES sobre o formato do Arquivo Texto: "dict.txt"

- O arquivo não possui letras acentuadas (caracteres especiais "fora" do ASCII padrão);
- Pode conter letras maiúsculas/minúsculas (mas são diferenciadas na tradução, ou seja, TABLE e Table e table são 3 palavras diferentes). Em resumo: não se preocupe em tratar isso.
- As palavras (e termos) não tem espaços, ou seja, dá para usar o fscanf ou fgets sem problema. Será usado o caractere de sublinhado '_' no lugar dos espaços. Exemplo, o termo "Artificial_Inteligente" aparece com '_' e é traduzido para "Inteligencia_Artificial".

  Sem espaços, sem acentos, e na tradução usamos a palavra exatamente como está (se tem minúsculas e/ou maiúsculas, deixa como está, e busca "como foi escrita").
- O separador dos campos é a vírgula ',' (como nos arquivos CSV e sempre sem espaços em branco).
- As linhas terminam com um '#' (evita problemas com \n, \r, \0 e outros caracteres invisíveis).
- A linha final do arquivo contém apenas um '#' (fica mais fácil de identificar o final do arquivo).
- As vírgulas devem ser "descartadas" e no lugar delas é colocado um '\0' (fim de palavra/string).
- Só vamos usar caracteres "ASCII normais" => do 33 '!' ao 125 '}'
  Não usaremos caracteres especiais, ou espaços, ou letras acentuadas no texto.
- Pode usar fscanf ou fgets para ler o arquivo.
  Tamanho máximo de uma linha de texto: 100 caracteres.
  Tamanho máximo de uma palavra/termo no texto: 45 caracteres.

➢ **Dicionário**: sempre tem 2 palavras, a 1a. em inglês, seguida da 2a. em português, separadas por uma vírgula, e terminada a linha por ',' e '#' (",#") no final.

➢ A última linha do arquivo tem uma '#' como primeiro caractere da linha de texto.

(2) Criação de duas árvores de dicionário para busca de palavras e tradução. A primeira Árvore Binária Ordenada é criada ordenando as palavras em Inglês (insere ordenado pela primeira palavra do arquivo do dicionário, que está em inglês). Isso vai permitir uma busca binária (rápida) pela palavra em inglês. A segunda Árvore tem a mesma estrutura, mesmo tipo de nodos (palavra em US "inglês" e palavra em PT "Português"), porém é ordenada pelas palavras em Português (insere ordenado pela segunda palavra do arquivo do dicionário, que está em português).

Note que o nodo deve conter o par de palavras usadas da tradução, por exemplo, armazena "book" e "livro", o que vai permitir traduzir de inglês para português (busca na árvore ordenada em inglês) e de português para inglês (busca na árvore ordenada em português).

NOTA IMPORTANTE: Se você não conseguir fazer as DUAS árvores, faça apenas uma, usando a ordenação usando as palavras em inglês, e traduzindo apenas de Inglês para Português (USPT). Mesmo não tendo a tradução inversa (implementação parcial), a sua prova será bem avaliada, O importante é ter pelo menos um tradutor de inglês funcionando.

(3) Consulta ao dicionário e tradução de palavras. Uma vez construídas as árvores que armazenam o dicionário, passamos para a fase de consulta ao dicionário. A consulta vai ser realizada digitando palavras pelo teclado (scanf) e exibindo na tela a tradução (printf) conforme a opção do modo de tradução escolhido (USPT: de Inglês para Português, ou, PTUS: de Português para Inglês). Procure implementar PELO MENOS a tradução USPT. Os dados entrados pelo teclado, e que estarão disponíveis nos casos de teste, seguem o seguinte formato: palavra, virgula em cada linha e termina com uma '#' (linha que indica o final da tradução e término da execução do programa).

<Tipo_de_tradução_USPT_ou_PTUS>,

<Palavra_ou_Termo>,

... mais palavras a serem traduzidas seguidas pela vírgula ...

#

Exemplo de interação com o programa implementado na prova:

Entrada (Teclado) Saída (Tela) Traduz de Inglês para Português (USPT)

```
USPT,
the,                o+a
book,               livro
is,                 esta
on,                 sobre
the,                o+a
table,              mesa
#                   #
```

NOTA: Veja que na tela, cada linha tem um "enter" - '\n' (sem vírgula), e também é exibido o '#' final.

Mais um exemplo de interação com o programa implementado na prova, agora com a outra tradução
Entrada (Teclado) Saída (Tela) Traduz de Português para Inglês (PTUS)

```
PTUS,
o+a,                the
livro,              book
esta,               is
sobre,              on
o+a,                the
mesa,               table
#                   #
```

NOTA: veja que não usamos acentos nem espaços em branco nos textos lidos.

Algumas dicas...

STRINGS
Lembre-se que você está usando TEXTOs, onde é importante usar a biblioteca "string.h"
Lembre-se que strings são copiadas com a "strcpy" (não pode atribuir direto!)
Lembre-se que a comparação de strings deve ser feita com a função "strcmp" e esta função retorna:
0 se as strings são exatamente iguais; e um valor < 0 ou > 0 se uma string "antecede" ou "sucede" a outra na ordenação. É como com a ordem numérica, mas considerando strings.
https://www.man7.org/linux/man-pages/man3/strcmp.3.html (menor, igual ou maior na comparação)

Cuidado com a diferença de final de string no CodeBlocks em relação ao RunCodes, foi por isso que os arquivos usam delimitadores bem definidos (as vírgulas) e os caracteres de "#", para não ter problema com o final da linha lida. Por isso é necessário "isolar" as palavras para usar no programa apenas a palavra, sem delimitadores ou marcas de final de linha, só a palavra em si.

Para poder traduzir de US para PT (USPT) e de PT para US (PTUS) será necessário criar duas árvores de busca, uma para buscar US e assim achar a palavra associada a palavra buscada em US, e uma outra árvore de busca, ordenada em PT para buscar PT e assim achar a palavra associada a palavra buscada em PT. São duas árvores, mas possuem a mesma estrutura, só muda a escolha da palavra usada para a ordenação. E atenção na parte final, tudo começa com a digitação de qual é a tradução que será realizada durante (toda) a interação com o usuário, se USPT ou se PTUS (primeira letra 'U' ou 'P').

* É "saudável" avisar se não conseguiu abrir e ler o arquivo de entrada e também se não conseguiu achar uma palavra ao buscar a tradução dela na árvore... ajuda e ver se o seu programa tem problemas.

BOA PROVA!!!

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É **INDIVIDUAL com consulta Papel (Livros, papel impresso ou escrito) e formato Digital (Internet, Wiki, Pendrive), porém SEM CONSULTAR ou SE COMUNICAR COM HUMANOS ou QUALQUER OUTRA FORMA DE VIDA TERRESTRE ou EXTRA-TERRESTRE!**
   Não podem usar de formas de comunicação com pessoas externas, além do professor, referente a prova, seja por celular (manter desligado/guardado), por e-mail, por whatsapp, por mensagens ou fóruns ("ao vivo"), com colegas, etc.
   NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL (Cadernos, Anotações, Livros, PenDrive, etc).

2. **A PROVA TEM QUE SER FEITA NO LABORATÓRIO 8.113 / 8.115 USANDO OS COMPUTADORES PCs DO LABORATÓRIO (SEM VPN, SEM ACESSO REMOTO!)**
   O ENDEREÇO DE IP USADO (Endereço de rede) PARA O ENVIO DA PROVA TEM QUE SER DAS MÁQUINAS DO LABORATÓRIO 8.113/8.115
   >> As máquinas do Lab. 8.113 e 8.115 estão sendo monitoradas << Sniff, sniff 😉

3. A PROVA DEVE SER **ENTREGUE via RunCodes.icmc (pode submeter várias vezes)**
   ENVIAR O ARQUIVO ZIP do MAKEFILE – O envio de ".c" é "aceito", mas tem *desconto* na nota.
   Entrega é via https://runcodes.icmc.usp.br/ Disciplina SSC0603 – Prova P2 Final

4. **EM CASO DE PROBLEMAS no RunCodes.icmc pode ser entregue por E-Mail** para:
   fosorio @ icmc.usp.br com **cópia** (**Cc:**) para fosorio @ gmail.com
   Com assunto/subject: **Prova-P2 <Seu_Nome> <Nro_USP>**
   ANEXANDO O PROGRAMA FONTE (Zip do Makefile ou se não conseguir, o ".C" e o Project ".CBP")
   NÃO ANEXAR EXECUTÁVEIS! JAMAIS ENVIE EXECUTÁVEIS POR E-MAIL
   NÃO ANEXAR OBJ, BIN e EXE!!! Seu e-mail não será recebido se for anexado bin, obj ou exe!

5. A PROVA USUALMENTE DEVE SER REALIZADA USANDO O CODEBLOCKS DO LABORATÓRIO, e depois enviada (testada com os casos de teste) do RunCodes (USUAL).

6. **AO ENTREGAR A PROVA NO RUNCODES ou POR E-MAIL O(A) ALUNO(A) CONCORDA COM ESTAS REGRAS E SE COMPROMETE A FAZER A PROVA INDIVIDUALMENTE!**

FIM