

Avaliação 3: Trabalho de Tabela de Hashing (Hash Map)

Obrigatório: para todas as soluções, construa a área de typedef contendo o projeto das estruturas de dados mencionadas nos enunciados e/ou criadas por você.

Não utilize comandos em C diferentes daqueles abordados em aula.

Total: 30 pontos.

Modalidade: trabalho em dupla.

Entrega: 27/06/2023, 23h55, via ava.

Questão Única

Implementação do tad hash map: 15 PTS

A implementação do tad hash map é aquela desenvolvida em aula e baseada no tad lista.

A interface do tad hash map está definida no arquivo tadh.h fornecido como material da avaliação. Este arquivo não poderá ser alterado. Todas as funções mencionadas no arquivo tadh.h devem ser implementadas com base no tad lista. Todas as estruturas e tipos de dados necessários à construção do tad hash map estão definidos nos arquivos tadh.h e tadlista.h.

Aplicação do tad hash map (descrição): 15 PTS

Um arquivo chamado di-pt-in.txt representa um dicionário de idioma (di) contendo uma lista de 250 pares de palavras português(br)-inglês(us). Arquivos equivalentes para os idiomas espanhol e francês estão disponíveis nos arquivos di-pt-es.txt e di-pt-fr.txt, respectivamente. Neste contexto, português é chamado de idioma origem e, inglês, espanhol e francês são chamados de idiomas destinos.

Utilizando **apenas** strings dinâmicas de tamanho igual a 64 caracteres para armazenar chaves e conteúdos, construa o que é solicitado:

Crie um arquivo de aplicação C chamada **aphm.c** contendo as funções listadas nos itens a seguir:

a) **t_hashmap carregadi(char *nomearq):** a função carrega o conteúdo de um dicionário de idioma cujo nome é passado como parâmetro de entrada. O arquivo deve ser processado linha a linha e com detecção de fim de arquivo via feof(FILE *). Como valor de retorno, a função devolve um dicionário do tipo t_hashmap cuja chave é uma string contendo a palavra origem e o valor é uma string contendo a palavra no idioma destino. **3 PTS**

b) **t_hashmap invertedi(t_hashmap paramdi):** a função recebe como entrada um dicionário de idiomas e devolve outro dicionário de idiomas com chave igual à palavra destino do dicionário de entrada e conteúdo igual à palavra de origem no dicionário de entrada. Exemplo: entrada é dicionário português-inglês, saída aé um dicionário inglês-português. **3 PTS**

c) **thash_map combinadi(t_hashmap diA, t_hashmap diB):** retorna um dicionário de idiomas que é a combinação dos dicionários de idiomas de entrada. O dicionário devolvido possui como chave a palavra destino de diA e como conteúdo associado à chave a palavra destino de diB. Exemplo: a entrada são os dicionários pt-en e pt-fr, a saída será o dicionário en-fr. **3 PTS**

d) **thash_map salvadi(t_hashmap di, char *nomearq):** a função recebe um argumento do tipo t_hashmap e uma string contendo um nome de arquivo. A função salva o conteúdo de di no arquivo com o mesmo fomrato usado nos arquivos de idioma descritos no primeiro parágrafo. Ao final, a função retorna o t_hashmap de entrada. **3 PTS**

e) **int main():** a função deve: **i)** carregar os arquivos de idioma di-pt-in.txt, di-pt-es.txt e di-pt-fr.txt e criar as respctivas variáveis tipo t_hashmap; **ii)** gerar os novos dicionários de idioma (t_hashmap) inglês-português, francês-português, espanhol-português, inglês-francês, espanhol-francês, inglês-francês; **iii)** cadastrar os novos dicionários no dicionário de idioma master (ver descrição no anexo 1); **iv)** interagir com o usuário para produzir a aplicação descrita no anexo 2. **3 PTS**

Anexo 1: Estrutura do dicionário master

O dicionário master é um dicionário de dicionários de idiomas. A chave é uma string do tipo <linguagem origem>-<linguagem destino> e o valor associado à chave é um dicionário de idiomas t_hashmap.

Exemplo de um dicionário python com a estrutura do dicionário master (dm):

```
dm = {  
    "pt-en": {"casa": "house", "prego": "nail", "água": "water", "torta": "pie", etc..},  
    "pt-es": {"sua": "su", "mas": "pero", "foram": "fueron", "você": "usted", etc..},  
    "pt-fr": {equivalente francês das chaves acima},  
    ...  
}
```

Anexo 2: Aplicação ED Translator

```
-----  
ED TRANSLATOR VERSÃO 1.0  
  
TRADUZIR DO IDIOMA (pt,en,fr,es): en  
PARA O IDIOMA (pt,en,fr,es): pt  
ENTRE COM A PALAVRA A TRADUZIR: horse  
TRADUÇÃO: cavalo  
-----
```

Os valores à frente dos campos de entrada de dados são apenas exemplos.

Após fornecer os idiomas de entrada e saída, a aplicação deve produzir a chave de pesquisa a ser usada no dicionário master e selecionar o dicionário de idiomas que fará a tradução da palavra desejada.

Caso a palavra fornecida para a tradução não seja encontrada, emitir uma mensagem de PALAVRA NÃO ENCONTRADA! e retornar à tela principal.

Para sair do loop do sistema, digitar a palavra FIM como dado de entrada no campo TRADUZIR DO IDIOMA.

Entrega

Compacte os arquivos aphm.c, tadh.c, tadh.h, tadlista.c, tadlista.h, di-pt-es.txt, di-pt-fr.txt e di-pt-en.txt em um arquivo ponto zip chamado avalhm.zip.

Compactações em formato rar não serão corrigidas.

Entregue o arquivo compactado na tarefa do ava aberta para a avaliação do tad hash map.

Aviso importante

Somente serão corrigidas as soluções entregues via tarefa do ava associada à avaliação. **Soluções não entregues, ou entregues via outros mecanismos ,receberão a nota zero.**

Caso não consiga fazer a entrega da tarefa devido a questões técnicas como falta de conexão com internet ou servidor ava fora do ar, registre a causa do impedimento e **procure o Professor no dia seguinte a entrega.** Compareça levando o arquivo avalhm.zip e providencie o devido envio para a tarefa do ava.

Não seguir o procedimento do parágrafo anterior implicará em nota zero para a avaliação.

Bons estudos!!