

USP – ICMC – SSC

SSC0603 – Estrutura de Dados I (ED1)

Professor responsável: *Fernando Santos Osório*

Semestre: 2021/2

Horário: Terça 16h20 e Quinta 14h20

Wiki: SSC-603 e **Facebook:** SSC0603-Fosorio

Web: <http://www.icmc.usp.br/~fosorio/>

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

DATA: 05 / 10 / 2021

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

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

PROVA P1 – SSC0603 ED1

Implemente o seguinte programa descrito abaixo de acordo com o especificado e USANDO UMA LISTA DINÂMICA ENCADEADA para armazenar os dados manipulados no programa. Pode usar LDES / Lista Simples, ou, LDED / Lista Dupla, o que achar melhor. Pode usar o TAD do Livro do Backes ou qualquer outro TAD de sua preferência. O programa deve ser bem “estruturado e organizado” (Uso de TAD, modular).

Autocompletar: O programa deve ler de um arquivo, com um conjunto de palavras e as estatísticas de uso destas palavras (contagem de vezes que foi usada), e assim implementar uma função de autocompletar, ou seja, sugere as palavras mais usadas para completar o texto que o usuário começou a digitar (baseado nesta “coleção de dados e estatísticas” armazenadas em uma lista. Exemplo:

Digitando “MA” poderia sugerir para autocompletar com: MACACO (33), MARIMBONDO (20), MARIPOSA (2), MARISCO (1), MARITACA (5), MARMOTA (12), MARRECO (7)

O programa vai sugerir apenas as 3 mais usadas: MACACO (33), MARIMBONDO (20), MARMOTA (12)

Detalhamento da Implementação:

O programa deve ler um arquivo texto (em disco) contendo conjuntos de dados no seguinte formato:

<prefixo> <palavra> <contagem> [Uma “tripla” destes dados por linha do arquivo]

Onde: <prefixo> é uma string de 2, 3 ou 4 letras. Exemplo: “MA”, “GOT”, “PALA”

<palavra> é uma string que representa uma palavra completa.

Exemplo: “MACACO”, “GOTA”, “PALAVRA”, “MAMAO”

Tanto o prefixo quanto a palavra são **sempre com letras maiúsculas e sem acentos ou caracteres especiais.**

<contagem> é um número, que indica o número de vezes que aquela palavra já foi usada.

Exemplo: (ver arquivo “arq.txt” e “arq1.txt” disponibilizados junto a prova)

MA MACACO 25

Leitura dos dados e criação da(s) lista(s):

1. LER arquivo, CRIAR lista e EXIBIR lista.

- Ler os dados dos arquivos, onde você pode usar fscanf “normal” (string, string e número), ou pode ler a linha com fgets e depois obter os pedaços com sscanf (string, string e número);
- Colocar os dados em uma lista encadeada, onde é fortemente aconselhado que seja usada uma lista do tipo LDED com os 3 campos de dados, ordenada pelo <prefixo> (1º. critério) e depois pela <contagem> (2º critério de ordenação).

ATENÇÃO:

>> Não esqueça que você vai estar manipulando STRINGS, e portanto tem que incluir o string.h e usar SEMPRE strcpy, strcmp, ... quando for manipular strings.

>> Copie a string para dentro do nodo com STRCPY pois você corre o risco de copiar ponteiros que depois vão se perder! Por exemplo: **strcpy (no->dados.txt, dt.txt);**

>> Lembrando: **strcmp** compara strings e indica se uma string é menor (anterior na ordenação), igual, ou, maior (posterior na ordenação)

- Depois de criada a lista, **exibir** na tela os dados da lista criada.

Usuário fornece <prefixo> e você consulta lista, mostrando as 3 melhores sugestões do autocompletar:

2. LER prefixo, CONSULTAR lista, EXIBIR sugestões (3 melhores, ou, se não achou o prefixo!).

- Usuário digita um prefixo e o programa procura por até 3 sugestões de palavras para autocompletar. As 3 sugestões são sempre as palavras mais usadas (com a maior <contagem> de uso).

• Exemplo da tela de execução do Programa:

```
>>> Autocompletar Palavras <<<
>> Lendo arquivos de dados: arq.txt
15 dados lidos
ARA ARANHA 15
ARAR ARARA 15
MA MACACO 33
MA MARIMBONDO 20
MA MARITACA 5
MA MARIPOSA 2
MA MARISCO 1
MAR MARMOTA 12
MAR MARRECO 7
TA TARTARUGA 15
TA TATU 13
TA TARANTULA 12
TA TAMANDUA 10
TA TAINHA 5
ZEB ZEBRA 1

>> Autocompletar:
Informe o prefixo: (2, 3 ou 4 caracteres)
Prefixo: MA
Sugestões:
MACACO 33
MARIMBONDO 20
MARITACA 5

Pressione qualquer tecla para continuar. . .
```

Funções de manipulação de Arquivos na linguagem "C":
fopen, fscanf, fgets, sscanf, strcmp, strcpy, ...
Use o devido #include e declaração de variáveis!

REGRAS EM RELAÇÃO REALIZAÇÃO DESTA PROVA

1. A PROVA É **INDIVIDUAL**.
2. PROVA DEVE SER REALIZADA CONSIDERANDO
AS REGRAS DO "Contrato" QUE VOCÊ ACEITOU => <https://forms.gle/fiiGFndiFA8MMgV27>



Autocompletar...

