

USP – ICMC – SSC

SSC0603 – Estrutura de Dados I (ED1)

Professor responsável: *Fernando Santos Osório*

Semestre: 2023/2

Horário: Terça 14h20 e Quinta 14h20

Wiki: SSC-603-2023 e **Facebook:** SSC0603-FOsorio

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

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

DATA: 19 / 09 / 2023

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

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

>> O PROGRAMA DEVE SER ENVIADO PARA O RUNCODES.ICMC com NRO USP NO CÓDIGO FONTE

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).

Lista de Valores FLOAT: O programa deve ler do teclado uma lista de valores FLOAT que serão armazenados de forma ordenada em uma lista encadeada. TODOS os valores lidos devem ser armazenados na lista, onde qualquer remoção ou alteração da lista de dados lidos será feita DEPOIS de ter inserido TODOS os dados na lista, ou seja, TODO dado digitado deve ser armazenado em memória na lista encadeada para depois ser processado conforme especificado. A lista termina por um valor “especial”, que não será armazenado na lista, sendo este valor um número negativo menor que -999.99. Exemplo:

Digitando como entrada do programa, como por exemplo:

10.0
2.99
3.77
10.0
9.20
-1000.00

1. Criar uma lista encadeada dinâmica e ordenada como esta: (sempre exibir com 2 casas decimais os valores)

Início => 2.99 => 3.77 => 9.20 => 10.00 => 10.00 => NULL

(*) Note que não foi inserido na lista o valor que identifica o final da lista de entrada (-1000.00)

2. **Exibir a lista criada na saída**, com os dados “printados” em ordem crescente na tela: cada valor deve ser exibido na tela com apenas 2 casas após a vírgula e com um “enter” ('\n') após cada valor. Exemplo:

2.99
3.77
9.20
10.00
10.00

3. Uma vez criada a lista, **devem ser removidos os números duplicados desta lista** e deve ser **ordenada do maior para o menor valor**, podendo usar rotinas de remoção, de criação de uma nova lista com inserção ordenada diferente, alterando o percurso da lista, usando outros ponteiros (cursor), como você achar melhor!

ATENÇÃO: serão considerados números duplicados aqueles que forem iguais até a 2a. casa após a vírgula, por exemplo: 10.001 é considerado igual a 10.000 e 1.1234 é considerado igual a 1.124 ou 1.122 (1.12). Atenção ao comparar números FLOATs pois já vimos que números de ponto flutuante devem sempre ser considerados levando em consideração uma certa “precisão” (número de casas), pois pode haver um erro (muito pequeno) na representação de ponto flutuante. DUAS casas após a vírgula são suficientes para considerar iguais, e 3 casas podem ser suficientes na comparação.

ATENÇÃO: se houver um número duplicado, ou triplicado, ou repetido mais vezes, APENAS UM VALOR é preservado na lista, os demais são removidos (não aparecem na lista final ordenada do maior para o menor). No exemplo dado, um nodo com o valor 10.00 deve ser removido, e um nodo com o valor 10.00 deve permanecer na lista. Se houvessem 3 valores 10.00, somente um valor 10.00 deveria permanecer na lista final.

Exemplo da Lista Encadeada Ordenada final:

Início => 10.00 => 9.20 => 3.77 => 2.99 => NULL

4. **Exibir a lista final criada na saída**, com os dados “printados” em ordem crescente na tela: cada valor deve ser exibido na tela com apenas 2 casas após a vírgula e com um “enter” ('\n') após cada valor. Exemplo:

10.00
9.20
3.77
2.99

5. FIM da execução do programa. Terminar o programa.

ENTREGA DA PROVA:

- A prova deve ser enviada no **RunCodes.ICMC** (site: <https://runcodes.icmc.usp.br/> no seu nome)
Na turma SSC0603 Estrutura de Dados I - 2023-2 (nossa turma que estamos usando atualmente)
- O Exercício é o “PROVA P1 – ED1 2023 – Fosorio
- Enviar um programa fonte único (programa .c) OU um arquivo .ZIP com Makefile e os fontes no padrão aceito pelo RunCodes.
- Não esqueça de colocar o seu nome como comentário nos programas fontes
- Se der problema com o envio para o RunCodes (**apenas se não conseguir enviar**), avise o professor e envie por e-mail para: fosorio@icmc.usp.br e com cópia para fosorio@gmail.com

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, com colegas, etc.

NÃO É PERMITIDO O EMPRÉSTIMO DE MATERIAL (Cadernos, Anotações, Livros, etc).

2. **A PROVA DEVE SER FEITA NO COMPUTADOR DO LABORATÓRIO.**

Não é permitido uso de celular, notebook ou tablet na prova, USE APENAS O DESKTOP do LAB.

3. **A PROVA DEVE SER ENTREGUE ENVIANDO NO RunCodes.ICMC**

Ver detalhes sobre a “ENTREGA DA PROVA” na página anterior.

4. A PROVA USUALMENTE DEVE SER REALIZADA USANDO O CODEBLOCKS DO LABORATÓRIO, se for usar outro Compilador/IDE, **INDIQUE NO E-MAIL qual AMBIENTE, COMPILADOR e IDE usou!**

5. **AO ENTREGAR A PROVA O(A) ALUNO(A) CONCORDA COM ESTAS REGRAS E SE COMPROMETE A FAZER A PROVA INDIVIDUALMENTE!**

FIM
