

Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação Departamento de Ciências da Computação SCC0122 - Estruturas de Dados (2025)

Professora: Mirela Teixeira Cazzolato

Monitores: Lucas Lombardi Castro e Leonardo Rossi Luiz

Trabalho 07 - Agenda de Contatos com Listas Ordenadas

1. Introdução

Um grupo de estudantes decidiu criar uma aplicação simples, mas essencial: uma agenda para gerenciar contatos pessoais. O principal requisito do projeto é que a lista de contatos seja sempre exibida em ordem alfabética pelo nome, para facilitar a busca e a visualização.

Desenvolver um sistema que reordena a lista inteira a cada novo contato adicionado seria muito ineficiente. A solução ideal é utilizar uma estrutura de dados que insira cada novo contato diretamente em sua posição correta, mantendo a agenda constantemente ordenada sem esforço computacional extra.

A estrutura de dados ideal para essa funcionalidade é a **Lista Ordenada**, que garante a ordem dos elementos em todas as operações de inserção e remoção. Sua equipe foi contratada para desenvolver o protótipo desta agenda de contatos.

2. Descrição do Problema

O programa deverá ler uma sequência de comandos, um por linha, para manipular uma agenda de contatos. Cada contato terá um nome (string sem espaços), um telefone (string sem espaços) e um email (string sem espaços).

A lista de contatos deve ser mantida em ordem alfabética crescente de acordo com o nome. O programa deve continuar executando até que o comando sair seja inserido.

Objetivo	Comando	Descrição da Operação
Inserir Contato	inserir <nome></nome>	Lê os dados do contato e o insere na lista, mantendo a ordem alfabética pelo nome. Se um contato com o mesmo nome já existir, uma mensagem de erro deve ser exibida e a inserção não deve ocorrer. Caso contrário, nenhuma saída é impressa.
Remover Contato	remover <nome></nome>	Remove a primeira ocorrência do contato com o nome especificado. Se o contato for removido, imprime uma mensagem de confirmação. Se não for encontrado, informa o usuário.

Buscar Contato	buscar <nome></nome>	Busca pelo contato com o nome especificado. Se encontrado, exibe todos os seus dados. Se não, informa o usuário.
Listar Contatos	listar	Exibe todos os contatos na agenda, em ordem alfabética, sem removê-los. Se a agenda estiver vazia, informa o usuário.
Encerrar	sair	Encerra a execução do programa.

3. Especificações e Critério de Avaliação

Atenção! Para a avaliação do projeto, não será considerada apenas a nota do Run Codes, mas também o cumprimento das seguintes especificações:

- O código deve estar na linguagem C. É permitido o uso da biblioteca string.h.
- O código deve ser organizado usando um TAD para a **Lista Ordenada** e um TAD para o **Contato** (contendo nome, telefone e email).
- A lista deve ser alocada dinamicamente.
- A legibilidade do código é um fator importante. Use nomes de variáveis e funções que sejam claras e comente o código de forma apropriada.
- A função usuário (main()) não deve manipular diretamente a estrutura de dados. Para criar, modificar ou desalocar o deque, utilize apenas as funções fornecidas pelo TAD.

4. Exemplos de Entrada e Saída

Exemplo de Entrada	Exemplo de Saída
inserir Carlos 9999-8888 carlos@email.com inserir Ana 8888-7777 ana@email.com inserir Bruno 7777-6666 bruno@email.com listar	Contatos: - [Ana, 8888-7777, ana@email.com] - [Bruno, 7777-6666, bruno@email.com] - [Carlos, 9999-8888, carlos@email.com]
buscar Ana sair	Contato Encontrado: [Ana, 8888-7777, ana@email.com]

Exemplo de Entrada	Exemplo de Saída
inserir Maria 1111-2222 maria@email.com	
inserir Joao 3333-4444 joao@email.com	
buscar Pedro	Contato Pedro nao encontrado.
remover Maria	Contato Maria removido.
listar	Contatos:
	- [Joao, 3333-4444, joao@email.com]
remover Maria	Contato Maria nao encontrado.
sair	

5. Submissão

- 1. **Envie** seu código fonte para o run.codes. Para isso, compacte (.zip), os arquivos, junto com um documento **Makefile.** (Será executado o comando "make all" para compilar seu código, e o comando "make run" para executar o seu código)
- 2. **Tire dúvidas!** Se tiver dúvida sobre algo, mande um email para os monitores, ou participe da monitoria presencial.
- 3. É permitido o uso da biblioteca string.h. Para usar essa biblioteca, não é necessário adicionar flags no Makefile.