

# Primeiro Exercício de Implementação: Lista Estática de Inteiros

Implemente uma lista estática de inteiros em C. A lista deve ter um tamanho máximo predefinido. Implemente funções para adicionar, remover e buscar elementos.

## Adicionar

Insere um inteiro na lista.

## Remover

Remove um inteiro da lista.

## Buscar

Encontra um inteiro na lista.



# Detalhes e Requisitos do Exercício 1

- Definir um tamanho máximo para a lista (e.g., 10).
- Implementar as funções de adicionar, remover e buscar.
- Tratar erros de lista cheia ou vazia.

1

## Tamanho

Defina o tamanho máximo.

2

## Funções

Implemente as funções.

3

## Tratamento

Trate erros.



# Segundo Exercício de Implementação: Lista Dinâmica de Strings

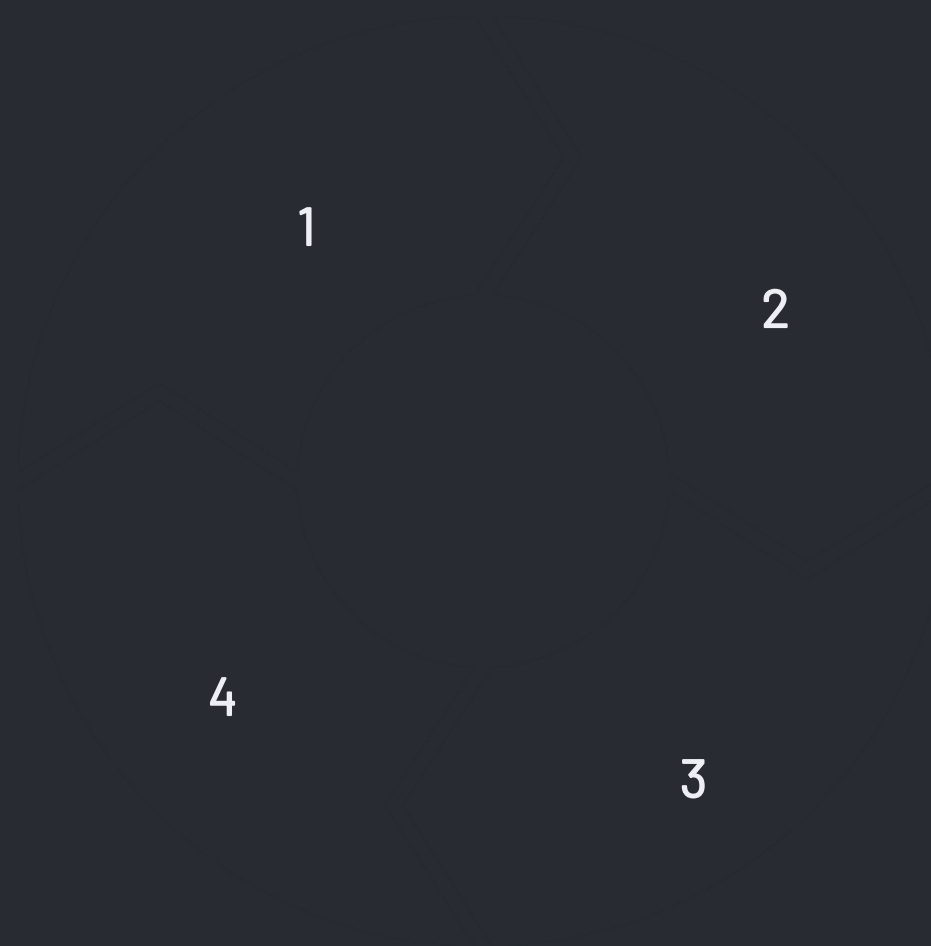
Implemente uma lista dinâmica de strings em C. Use alocação dinâmica de memória para redimensionar a lista conforme necessário. Implemente funções para adicionar, remover e buscar strings.

**Alocar**  
Alocar memória.

**Adicionar**  
Adicionar strings.

**Buscar**  
Buscar strings.

**Remover**  
Remover strings.



# Detalhes e Requisitos do Exercício 2

- Usar **malloc**, **calloc** e **free** para gerenciar a memória.
- Implementar funções para adicionar, remover e buscar strings.
- Tratar erros de alocação de memória.

## 1 Alocação

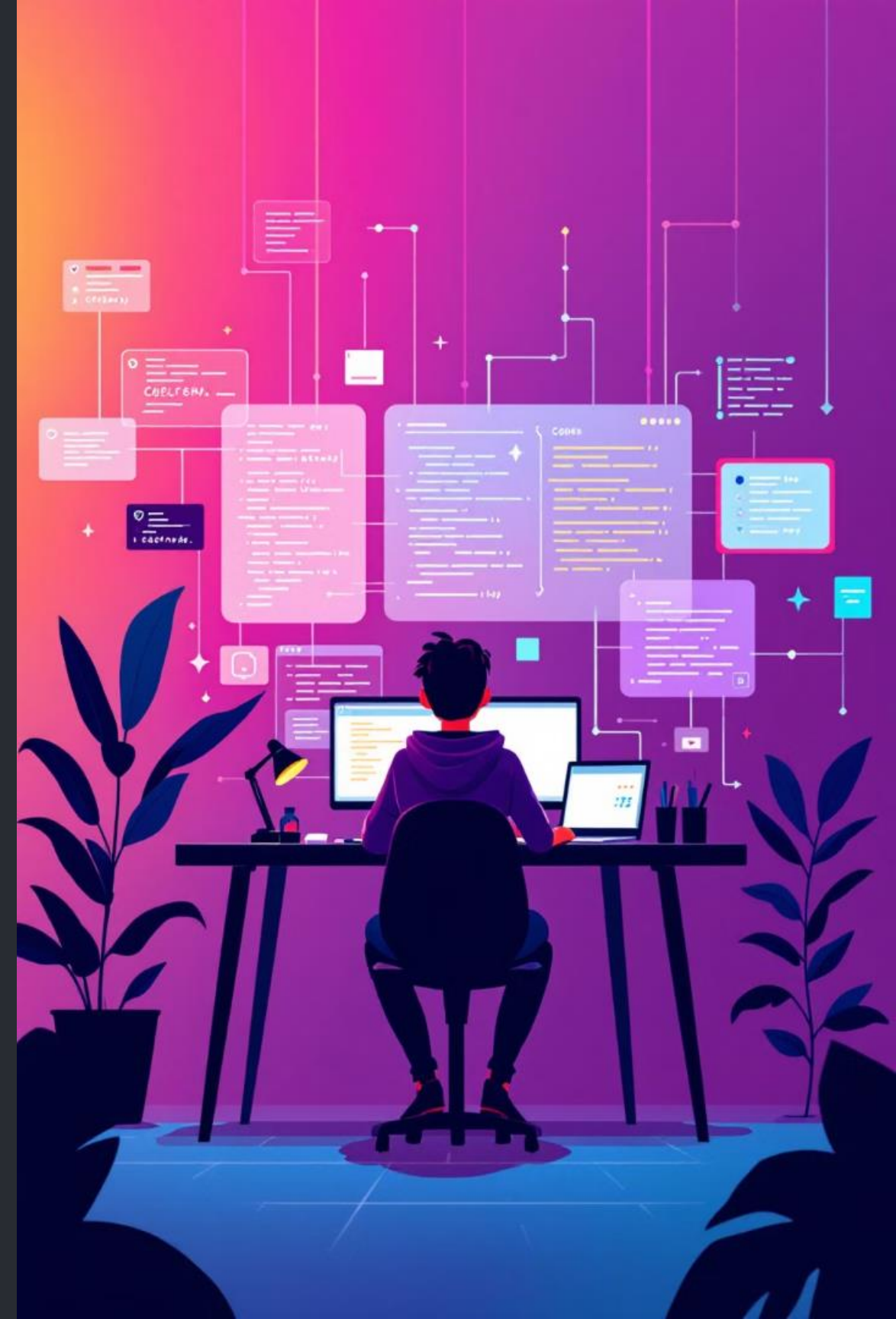
Use malloc/calloc.

## 2 Funções

Implemente funções CRUD.

## 3 Erros

Trate erros.







# Dicas para Implementação Eficiente

## 1 Evitar Vazamentos

Sempre liberar a memória alocada com **free**.

## 2 Alocação Otimizada

Alocar memória em blocos maiores para reduzir overhead.

## 3 Código Limpo

Usar nomes de variáveis e funções descritivos.

# Boas Práticas de Programação em C para Listas

## Modularização

Dividir o código em funções menores e reutilizáveis.

## Documentação

Comentar o código para explicar o funcionamento.

## Testes

Escrever testes unitários para garantir a correção.