

Curso	Ciência da Computação
Disciplina	Algoritmos e Estruturas de Dados 1
Professor	José Gustavo de Souza Paiva

1. Tema central

Definição e implementação de listas circulares na linguagem C

2. Objetivos

1. Construir implementações para listas encadeadas circulares;
2. Construir programas básicos para utilizar os conceitos aprendidos.

3. Informações

O aluno deverá resolver os problemas listados na seção 3. As resoluções deverão ser compactadas em um arquivo .zip ou .rar, e submetidas como resposta à tarefa correspondente no Microsoft Teams, até a data limite estipulada na tarefa. Não serão aceitas respostas enviadas por outros meios. Qualquer informação, dúvida ou sugestões quanto aos exercícios propostos neste material poderão ser encaminhadas ao professor através do e-mail: gustavo@ufu.br.

4. Conteúdo

Para cada um dos problemas abaixo, construa o programa completo na linguagem de programação C capaz de verificar a funcionalidade de cada uma das funções solicitadas. Utilize a representação de listas duplamente encadeadas vista em sala de aula para organizar os dados do programa.

- a) Implemente as funcionalidades previstas para o TAD de Lista Circular, considerando a abordagem de armazenamento do campo **FINAL** ao invés do campo **INICIO**, como discutido em sala de aula.
- b) Dada uma lista encadeada circular **l**, faça uma função que a receba como parâmetro, e remova o elemento na posição central. Por exemplo, caso a lista tenha 5 elementos, remova o 3º, caso a lista tenha 8 elementos, remova o 4º, e assim por diante.
- c) Faça uma função que, dado uma chave (matrícula do aluno), remova da lista a primeira ocorrência de um aluno com essa chave. Utilize essa função para implementar a função `int removerItem(Lista *l, int it)` presente no TAD `Lista.h`.
- d) Dadas duas listas **L1** e **L2**, faça uma função que crie uma outra lista contendo a interseção de **L1** e **L2**, retornando essa nova lista.
- e) Implemente uma função de inserção no final de uma lista circular que garanta a não repetição dos dados dessa lista. Assim, caso o elemento já exista na lista, a inserção não ocorre. A função deverá utilizar uma outra função chamada `contaOcorrencias`, que conta quantas ocorrências de um determinado elemento (passado como parâmetro) existem na lista.