



Lista de Exercícios

Aluno(a): Sthéfane dos Reis Almeida
Matrícula: 2021200533

Obs: Não consegui fazer o exercícios 07, tenho mais dificuldade com a lista duplamente encadeada, se puder fazer a correção dos exercícios em sala de aula eu agradeceria demais, vai ajudar a termos maior confiança em relação ao conteúdo

1. Crie um programa utilizando lista estática que simule uma agenda telefônica.
Para cada pessoa, deve-se ter os seguintes dados:
 - Nome
 - E-mail
 - TelefoneO programa deve utilizar lista estática para armazenar até 10 entradas.
Após a leitura, liste na tela os registros existentes.
2. Crie um programa que receba a matrícula, nome e as notas de 4 provas de 1 aluno. Use Struct para criar um TAD Aluno e crie uma função para calcular e retornar a média.
3. Num processo seletivo para admissão de alunos em uma escola e precisa cadastrar os dados dos candidatos. Para isso, construir um programa que armazene o nome, idade e cidade onde reside cada candidato. Armazene essas informações em uma estrutura de dados e utilize alocação dinâmica de memória para inserir cada candidato informado pelo usuário.
4. Considerando listas estáticas, faça uma função em C que verifique se a lista está ordenada.
5. Para listas estáticas, implemente uma função que elimine de uma determinada lista, todas as ocorrências de um dado elemento. Considere que a lista está ordenada.
6. Em que circunstâncias seria preferível utilizar a implementação estática ao invés da dinâmica para listas? Cite algumas vantagens e desvantagens que podem contribuir para esta justificativa.
7. Implemente as seguintes funções para a estrutura de dados lista duplamente encadeada vista em sala de aula:
 - (a) Uma função que retorna um ponteiro para o último elemento da lista, com o protótipo:
`Lista* ultimo(Lista *li);`
 - (b) Uma função que retorna o número de nós da lista que possuem o campo matrícula com valores maiores do que "n", com protótipo:
`int maiores(Lista* li, int n);`
 - (c) Função que concatena duas listas l1 e l2. Essa função deve retornar a lista l1 contendo todos os seus elementos (na mesma ordem) e, em seguida, todos os elementos de l2. A função deve seguir o protótipo abaixo:
`void concatena(Lista* l1, Lista *l2);`
 - (d) Função que receba como parâmetro uma lista e um valor inteiro n e divida a lista em duas,

de forma que a segunda lista comece no primeiro nó logo após a primeira ocorrência da matrícula igual a n na lista original. A figura a seguir ilustra essa separação.