



Centro de Ciências Exatas, Arquitetura e Engenharia

Professor:	Ciro Cirne Trindade
Disciplina:	Estruturas de Dados-I
Cursos:	Ciência da Computação/Sistemas de Informação
Data de entrega:	09/06/2021
Grupo:	Até 4 alunos

2º Trabalho Prático – Lista Duplamente Encadeada

Implemente um programa para manipular uma lista de alunos através de uma lista duplamente encadeada. Os dados dos alunos devem ser lidos de um arquivo binário em memória secundária (disco) de estruturas do tipo `taluno`, mostrada a seguir, e carregados para a lista em memória principal. A partir daí todas as operações devem ser feitas na lista.

A definição do tipo de dado armazenado na lista é dada abaixo:

```
typedef enum {  
    CIENCIA_DA_COMPUTACAO,  
    SISTEMAS_DE_INFORMACAO } tcurso;  
  
typedef struct {  
    int mat; // matrícula do aluno  
    char nome[61]; // nome do aluno  
    tcurso curso; // nome do curso  
    int idade; // idade do aluno  
} taluno;  
  
typedef struct lst_no * lst_ptr;  
struct lst_no {  
    lst_ptr ant;  
    taluno dado;  
    lst_ptr prox;  
};
```

Seu programa deve ser baseado em um menu e permitir as seguintes operações sobre a lista após ela ter sido carregada em memória principal:

- Incluir um novo aluno: o programa deve solicitar o número de matrícula, o nome, curso e a idade do aluno e incluir na lista. O programa não deve permitir o cadastro de dois ou mais alunos com o mesmo número de matrícula;
- Remover um aluno: o programa deve solicitar o número de matrícula do aluno, caso exista um aluno com a matrícula informada, exibir seus dados, e pedir uma confirmação da exclusão. Caso a matrícula não seja encontrada, o programa deve exibir uma mensagem de erro;

- c) Procurar um aluno: o programa deve solicitar o nome do aluno e listar todos os alunos cujo nome **comece** com o nome informado pelo usuário;
- d) Listar os alunos de um curso em ordem alfabética crescente: o programa deve solicitar o curso e listar, no formato de uma tabela, todos os alunos deste curso em ordem alfabética crescente;
- e) Listar os alunos de um curso em ordem alfabética decrescente: o programa deve solicitar o curso e listar, no formato de uma tabela, todos os alunos deste curso em ordem alfabética decrescente.

Ao sair do programa, você deve gravar a lista da memória principal para o arquivo binário em memória secundária.

Observações:

1. Critérios de avaliação:
 - a) Corretude: 80%
 - b) Interface: 10%
 - c) Legibilidade: 10%
2. Todos os trabalhos devem possuir no cabeçalho a identificação dos autores.
3. Trabalhos copiados (com ou sem eventuais disfarces) terão a nota dividida pelo número de cópias (inclusive o original).
4. Trabalhos atrasados terão um desconto de 1 ponto por dia de atraso até o limite de 5 dias de atraso.
5. Trabalhos com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
6. É muito importante que seu programa esteja bem comentado e bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. A legibilidade do código será levada em consideração pelo critério de avaliação do trabalho.
7. Você deve enviar os arquivos fontes (.c e .h), além do *makefile*, em um arquivo compactado através da funcionalidade de atividades práticas do sistema acadêmico.