IFPB – CAMPUS CAMPINA GRANDE.
CURSO: ENGENHARIA DE COMPUTAÇÃO
DISCIPLINA: ESTRUTURA DE DADOS/LABORATÓRIO DE ESTRUTURA DE DADOS

Atividade de Avaliação 2

(Questão): Utilizando técnicas de modularização vistas em sala, implemente o TAD **TListaAlunos** (de implementação **simplesmente encadeada**) que representa uma **lista de Alunos**. Implemente este tipo numa biblioteca chamada **listadinalu**. Esta biblioteca deve seguir o mesmo padrão utilizado nas aulas. Ou seja, contendo uma interface bem definida (definição de tipos e funções) em um arquivo com extensão **.h**, chamado **listadinalu.h**. O arquivo de implementação deve ser chamado **listadinalu.c**. A biblioteca deve possuir as seguintes funções:

Operações básicas:

a) Criação da lista simplesmente encadeada vazia
b) Verificar se a lista está vazia
c) Obter o tamanho da lista
d) Obter os dados do Aluno informando a posição
e) Obter a posição do Aluno quando a matrícula do Aluno é informada
f) Inserir um novo Aluno no final da lista
g) Remover um Aluno informando a matrícula
h) Exibir a lista com todos os alunos e seus dados em um formato de apresentação "amigável"

Além das operações básicas, implemente também as seguintes operações:

a) Inserir um Aluno na primeira posição da lista;
b) Inserir um novo Aluno dada uma posição
c) Remover um Aluno de uma determinada posição;
d) Pesquisar um aluno pela matrícula ou pelo nome (o usuário deve escolher o tipo de busca), e exibir todos os dados do aluno encontrado. No caso da busca pelo nome, se existir mais de um aluno, exiba as informações sobre todos eles.

Lembre-se:

- Em cada uma das operações, identifique possíveis situações de erros do usuário e exiba mensagens para ele nestas situações. (Ex. o programa deve exibir mensagens no caso do usuário tentar remover um Aluno numa lista que está vazia etc.);
- No programa principal (main.c), crie um menu de operações com todas as operações da lista descritas anteriormente.

Na definição do tipo que represente o Aluno (**TAluno**), faça da seguinte forma:

```
typedef struct aluno{
    int matricula;
    char nome[30];
    float nota1, nota2;
} TAluno;
```

Na definição do tipo lista de alunos (**TListaAluno**), faça da seguinte forma:

```
typedef struct no{
   TAluno dado;
   struct no *prox;
}No;

typedef struct lista{
   No *inicio;
}TListaAluno;
```

As assinaturas das operações básicas do TAD **TListaAlunos** devem ter o seguinte formato:

```
void cria(TListaAluno *la); // Cria a lista vazia
int tamanho(TListaAluno *la); // Retorna o tamanho da lista
int vazia(TListaAluno *la);  // Retorna se a lista está vazia
int buscaPosMat(TListaAluno *la, int mat, int *pos); // Pesquisa posição pela matricula
int buscaAlunoPos(TListaAluno *la, int pos, TAluno *al); // Pesquisa aluno pela pela posição
int insereFinal(TListaAluno *la, TAluno al); // Insere aluno no final da lista
int removeAlunoMat(TListaAluno *la, int mat); // Remove aluno da lista passando a matricula
void exibirAlunos(TListaAluno *la);  // Exibe a Lista de alunos
```

As demais operações implementadas podem seguir assinaturas semelhantes.