



Aula de Exercícios de Prog. II – Lista Simplesmente Encadeada

1. Considere um tipo que representa o desempenho de um aluno em uma disciplina definido pela estrutura a seguir:

```
struct aluno {
    char nome[81];
    int mat;
    float media;
};
typedef struct aluno Aluno;
```

Considere ainda um tipo que representa um nó de uma lista encadeada de alunos:

```
struct elemento {
    Aluno* info;
    struct elemento* prox;
};
typedef struct elemento Elemento;
```

E considere um tipo que representa uma lista encadeada, contendo três campos: um ponteiro para o primeiro elemento da lista, um ponteiro para o último elemento da lista e um inteiro com a quantidade de elementos na lista:

```
struct lista {
    struct elemento* prim; /* ponteiro para primeiro elemento */
    struct elemento* ult; /* ponteiro para ultimo elemento */
    int quant; /* quantidade de elementos na lista */
};
typedef struct lista Lista;
```

- a) Escreva uma função para criar uma lista vazia. Dica: essa função deve alocar dinamicamente uma estrutura lista e inicializar seus campos de primeiro e último para NULL e a quantidade com zero:

```
Lista* cria_vazia(void);
```

- b) Escreva uma função para inserir um novo aluno no início de uma lista. O protótipo da função deve ser conforme abaixo. Lembre-se de atualizar os campos da estrutura lista corretamente (incrementar a quantidade e atualizar o primeiro, e último quando for o caso):

```
void insere_inicio (Lista* l, char* nome, int mat, float media);
```

- c) Escreva uma função para inserir um novo aluno na lista, mantendo os alunos em ordem decrescente de média na lista. O protótipo da função deve ser:

```
void insere_ordenado (Lista* l, char* nome, int mat, float media);
```

- d) Escreva uma função para imprimir os dados de uma lista de alunos. O protótipo da função deve ser:

```
void imprime (Lista* l);
```





- e) Escreva uma função recursiva para retornar a quantidade de alunos reprovados na turma, assumindo que a média para passar é 5. O protótipo da função deve ser:

```
int reprovados(Elemento* elem);
```

- f) Escreva um programa principal que teste suas funções.

