Laboratório de Listas

1) Implemente o TAD ListaSequencial como está no slide. Entenda o código e escreva **sem** utilizar copy+paste. Crie no programa principal rotinas que utilizam todos as funções do TAD

2) Altere o TAD ListaSequencial para que suporte listas de tamanhos variáveis, mas que ainda utilize vetores para armazenar os elementos da lista.

Ao criar a lista, passe como parâmetro um tamanho inicial máximo de elementos da lista

Ex: 100 → li = criar_lista(100);

Caso o usuário tente inserir mais elementos do que a lista suporta, o TAD deve automaticamente realocar mais memória, adicionando a mesma quantidade inicial de elementos. Ou seja, se ao criar o TAD o usuário alocou inicialmente 100 elementos, quando o usuário tentar inserir o 101º elemento o TAD deverá alocar antes mais 100 elementos, totalizando 200 elementos no máximo. Se o usuário posteriormente tentar inserir o 201º elemento, mais 100 elementos serão alocados, totalizando 300 no máximo.

Ao remover elementos a memória já previamente alocada deve ser mantida. No entanto, crie uma função chamada compactar_lista, que reduzir o tamanho do vetor da lista para o menor tamanho possível que seja múltiplo do tamanho inicial da lista.

Exemplo:

Lista com 1000 posições máximas

Atualmente com 560 elementos.

Tamanho inicial: 100

Ao reduzir a lista ficará terá no máximo 600 posições alocadas.

ceil(560/100)*100 → 600

obs:

A função *ceil()*, disponível no *#include <math.h>* faz o arredondamento de um valor para cima. Exemplos:

ceil(2.3) => 3
ceil(2.9) => 3
ceil(2.0) => 2