

```

/** UEG - Campus Trindade - Data: 06/06/2022
 * Disciplina: Estrutura de Dados I
 * Professor: Claudio Martins
 * Discente: Ricardo Hatsugai
 * Problema proposto: Crie um programa que faça uma lista dinamica estática,
 * em que receba valores no início e no final, e imprima na tela esta lista.*/

#include <stdio.h>
#include <stdlib.h>

/* Criação do struct do Nó */
typedef struct No {
    int valor;
    struct No *proximo;
} No;

/* Criação da struct do tamanho e ponteiros de início e fim */
typedef struct {
    No *inicio, *fim;
    int tam;
} Lista;

/* Inserção no início da lista */
void inserirInicio(Lista *lista, int valor){ // função void que não retorna
valor, e sim faz uma ação
    No *novo = (No*)malloc(sizeof(No)); // cria um novo nó
    novo->valor = valor; // (*novo).valor = valor;

    if (lista->inicio == NULL){ // se a lista for igual a vazio.
        novo->proximo = NULL; // ponteiro para a próximo valor da lista recebe
vazio.
        lista->inicio = novo; // primeira posição da lista, recebe vazio.
        lista->fim = novo; // fim da lista também recebe vazio.
    } else { // senão estiver igual a vazio.
        novo->proximo = lista->inicio; // próximo valor da lista, recebe o valor
que é vazio
        lista->inicio = novo; // a primeira posição da lista, recebe o valor da
variável "valor"
    }
    lista->tam++; // aqui acontece a expansão da lista em mais uma posição.
} // fim da função inserirInicio

/* Inserir no final da lista */
void inserirFim(Lista *lista, int valor) { // função void que não retorna valor,
e sim faz uma ação.
    No *novo = (No*)malloc(sizeof(No)); // cria um novo nó
    novo->valor= valor; // (*novo).valor = valor
    novo->proximo = NULL; // próxima posição da lista, recebe vazio.

    if(lista->inicio == NULL){ // se lista for igual a vazio.
        lista->inicio = novo; // posição inicial recebe valor
        lista->fim = novo; // posição fim, que é a mesma da inicial, recebe
valor.
    } else { // se a lista não estiver vazia.
        lista->fim->proximo = novo; // a posição final da lista, que é a
próxima, recebe o valor.
        lista->fim = novo; // o final da lista recebe novo valor.
    }
    lista->tam++; // aqui acontece a expansão da lista em mais uma posição.
} // fim da função inserirFim.

/* Função pra imprimir uma lista */
void imprimir(Lista *lista){ // função void que não retorna valor, mas imprime a
lista.

```

```

    No *inicio = lista->inicio;
    printf("Tamanho da lista: %d\n", lista->tam);
    while(inicio != NULL){
        printf("%d ", inicio->valor);
        inicio = inicio->proximo;
    }
    printf("\n\n\n");
}

// programa que utilizará as funções
int main(){
    Lista lista; // criação da lista
    int opcao, valor; // criação das variáveis inteiras "opcao" e "valor"

    // inicia a lista
    lista.inicio = NULL;
    lista.fim = NULL;
    lista.tam = 0;

    /* Criação de um menu para as opções */
    do {
        printf("1 - Inserir no inicio\n2 - Inserir no fim\n3 - Imprimir\n4 - Sair\n");
        scanf("%d", &opcao);
        switch(opcao){
            case 1:
                printf("Digite um valor a ser inserido:");
                scanf("%d", &valor);
                inserirInicio(&lista, valor);
                break;
            case 2:
                printf("Digite um valor a ser inserido: ");
                scanf("%d", &valor);
                inserirFim(&lista, valor);
                break;
            case 3:
                printf("\nLista: \n");
                imprimir(&lista);
                break;
            case 4:
                printf("Finalizando...\n");
                break;
            default:
                printf("Opcao invalida!\n\n");
        }
    } while(opcao != 4);

    return 0;
}

```