INSTITUTO FEDERAL DA PARAÍBA
CAMPUS CAMPINA GRANDE
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO
DISCIPLINA DE LABORATÓRIO DE ESTRUTURAS DE DADOS
PROF. VICTOR ANDRÉ PINHO DE OLIVEIRA

Lab. Estruturas de Dados

Atividade Prática 6 - Listas Duplamente Encadeadas

Instruções

Responda às questões abaixo. Pode usar este próprio documento. Questões práticas devem ser anexadas separadamente.

Legenda:

Quirrell: Questões mais simples e diretas

Olho-Tonto: Questões de nível intermediário

Dolores: Questões de nível mais difícil, exigentes

Voldemort: Questões com nível de exigência altíssimo, desafiadoras

Questões

1. **Quirrell** Considerando o código da Lista Duplamente Encadeada não Ordenada presente no material, crie:
    a. uma função inserir_ini que permite inserir um elemento no início da lista;

```c
void inserir_ini(int dado){
  struct sNODE *novo = (struct sNODE*) malloc(sizeof(struct sNODE));
  novo->dado = dado;
  novo->ant = NULL;
  novo->prox = NULL;

  if (!ini)
    ini = fim = novo; //verifico se a lista está vazia, se sim, defino o novo nó como início e fim
  else{               //a principio a mesma lógica usada para inserir ao final
    ini->ant = novo;   // entretanto, com a lista possuindo algum nó, ant vai ser desconectado do terra sendo conectado ao novo nó
    novo->prox = ini; //nesse novo nó prox deve apontar para o nó existente
    ini = novo; //logo, o novo nó implementado passará a ser o novo início
  }
    //mas n para por aí, anteriormente o nó inicial tinha o ant aterrado. dessa forma, devemos fazer o mesmo com o novo nó:
    ini->ant = NULL;
}

```

    b. uma função imprimir_reverso que imprime a lista de trás para frente.

2. **Quirrell** A forma como implementamos nossas Listas no material traz uma grande limitação: o nosso programa só pode manipular uma Lista por vez, e isso não é bom. Podemos resolver isso criando um registro LISTA, que contém as variáveis particulares necessárias para o controle de cada LISTA. Desse modo, basta adicionarmos um novo parâmetro às funções para que elas operem em cima da Lista passada como argumento. A partir do código abaixo, implemente uma versão melhorada de uma Lista Duplamente Encadeada de inteiros.

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

struct sNODE{
  int dado;
  struct sNODE *ant;
  struct sNODE *prox;
```

```
};

struct sLISTA{
  struct sNODE *ini, *fim;
};

typedef struct sLISTA LISTA;

void inicializar(LISTA *lst);
void apagar(LISTA *lst);

void inserir_ord(LISTA *lst, int dado);
void remover(LISTA *lst, int dado);
struct sNODE *buscar(LISTA *lst, int dado);

int obter(struct sNODE *node);
int tamanho(LISTA *lst);
void imprimir(LISTA *lst);
```

Note que, da forma como criamos o registro, cada LISTA terá seus próprios ponteiros ini e fim. Veja um exemplo de como criar e usar a LISTA:

```
int main(){
  LISTA lst;
  inicializar(&lst);

  inserir_ord(&lst, 100);
  imprimir(&lst);

  apagar(&lst);

  return 0;
}
```

Considere:

- a função **inicializar** apenas inicializa os ponteiros ini e fim para NULL.
- a função **apagar**, por sua vez, deverá desalocar todos os nós da lista. Não esqueça de atribuir NULL aos ponteiros ini e fim.
- as demais funções farão a mesma coisa conforme visto no material. Desta vez, no entanto, considerando o parâmetro LISTA *lst, que é passada como ponteiro para cada função.

Em essência, o código está praticamente pronto no material, à exceção da função inicializar. Você fará apenas as adequações necessárias para atender às novas especificações.

3. **Olho-Tonto** Crie uma Lista Duplamente Encadeada que armazena em cada nó o nome de um funcionário, sua ocupação na empresa e o seu salário. Considere o tamanho das strings nome e ocupação de, no máximo, 30 caracteres. A Lista deve inserir os funcionários já Ordenados considerando os salários (primeiro os maiores salários; por último os menores salários).

4. **Voldemort** Crie uma Lista Duplamente Encadeada que armazena em cada nó o nome de um funcionário, sua ocupação na empresa e o seu salário. Considere o tamanho da string nome de, no máximo, 30 caracteres. Considere também que a ocupação seja: Gerente ou Supervisor ou Peão. A Lista deve inserir os funcionários já Ordenados considerando primeiro a ocupação (primeiro os Gerentes, depois os Supervisores e, por último, os Peões) , e depois os salários (primeiro os Gerentes com maiores salários … por último os Peões com menores salários).