Lista Simplesmente Encadeada (*LSE*)

ATENÇÃO. O trabalho pode ser feito em **duplas**. Somente **UM dos integrantes da dupla deve submeter** o trabalho no *moodle*. Os nome do arquivo deve ser **Aluno01-Aluno02.zip**.

Problema para o laboratório. Nós vamos implementar uma lista simplesmente encadeada de números inteiros positivos em que os números pares devem ser inseridos no início da lista, os números ímpares devem ser inseridos no final da lista, e os zeros devem ser inseridos no meio da lista. Os valores não ficam ordenados. Isto significa que você deve ler uma quantidade indeterminada de números (o critério de parada é um número menor do que zero). Para cada número maior do que zero, verificar se o número é par ou ímpar ou zero e chamar a função de inserção para inserir o número lido ou no início ou no final ou no meio da lista. Veja os exemplos

Lista: 2 4 6 7 5 9
Insere: 0
Saida: 2 4 6 0 7 5 9

Lista: 2 4 9 7 5 3 1 15
Insere: 0
Saida: 2 4 9 7 0 5 3 1 15
Atenção: o zero fica no meio da lista e não entre os pares e impares.

Lista: 2 4 9 7 5
Insere: 0
Saida: 2 4 0 9 7 5
Atenção: se a lista tiver um número impares de nós, inserir imediatamente antes do meio da lista.

Note que para inserir um nó exatamente no meio da lista, você deve percorrer toda a lista para saber quantos nós a lista possui no momento, pois na lista encadeada temos apenas busca sequencial.

Resumo:

- Número par → insere no início da lista
- Número ímpar → insere no fim da lista
- Número = 0 → insere no meio da lista
- Número < 0 → encerra o laço

Os nós dessa lista são do tipo `numero`, definido como o tipo estruturado `tipoNumero` descrito a seguir. A função insere deve manter um único ponteiro (`ptLista`) para o início da lista.

```
struct numero
{
   int num;
   struct numero *prox;
};
typedef struct numero tipoNumero;
```

Passos para testar o laboratório (`main()`):

a) Criar lista vazia
b) Inserir os elementos na lista (número indeterminado de elementos, conforme descrição acima). Dividir o insere em três funções separadas, como a seguir:
    a. InserirIni alocar memória e inserir o nó no início da lista. Atenção, é preciso tratar os casos de lista vazia e lista não-vazia.
    b. InserirFim alocar memória e inserir o nó no final da lista. Nesse caso, você deve percorrer a lista até o final para poder inserir o novo nó. Atenção, é preciso tratar os casos de lista vazia e lista não-vazia.
    c. InserirMeio alocar memória e inserir o nó no meio da lista. Nesse caso, você deve percorrer a lista para contar quantos nós existem para poder encontrar o meio da lista. Se a lista tiver um número impar de nós, inserir após a metade. Atenção, é preciso tratar os casos de lista vazia, lista não-vazia e insere no inicio ou no fim.
c) Exibir todos os elementos da lista (do início ao fim)
d) Destruir a lista
e) Exibir todos os elementos da lista (do início ao fim)

Você pode aproveitar (se achar útil) a implementação disponível no moodle para aula LSE.