

Lista encadeada

Você pode utilizar qualquer ambiente de programação para desenvolver sua atividade. Ao final, **copie e cole o seu código-fonte com a resposta aqui mesmo neste documento**, dentro dos espaços indicados para isso e **preservando a indentação do código**. **Depois que terminar sua avaliação, não se esqueça de entregar sua atividade!** Fique atento ao relógio, pois as atividades entregues com atraso não serão aceitas.

Para resolver esta atividade, [clique aqui para baixar](#) o projeto da aula 7 de laboratório de programação 2, que contém a implementação do TAD ListaEncad para lista encadeada de números inteiros. Na sua solução para a questão abaixo, [você pode utilizar/chamar](#) qualquer uma das operações que estejam disponíveis no projeto (exatamente do jeito que ele se encontra no site da disciplina). Quaisquer outras operações necessárias para a sua resposta devem ser implementadas e incluídas neste documento.

Implementar a operação `void ListaEncad::reduz_n(int n1)`; para diminuir o valor de `n` para o novo valor `n1`, onde `n` é o número de nós de uma **lista simplesmente encadeada com descritor**. O parâmetro `n1` tem que estar entre 0 e `n`. Não basta apenas atribuir à `n` o seu novo valor `n1`, um número adequado de nós deve ser liberado para que a quantidade de nós alocados fique coerente com o novo valor para `n`. **Desalocar os últimos nós** da lista de forma que ela fique com os primeiros `n1` nós. Emitir a mensagem "Valor invalido" caso `n1` não esteja entre 0 e `n`. A operação `reduz_n` tem que ser $O(n)$.

Exemplo: Considere a lista simplesmente encadeada com descritor `L` com os valores `L = [36 41 26 15 45]`:

- O comando `L.reduz_n(2)` torna `L = [36 41]`.
- O comando `L.reduz_n(0)` torna `L = []` (lista vazia).

```
void ListaEncad::reduz_n(int n1)
{
    if (n1 < 0 || n1 > n) //Caso n1 for inválido
    {
        cout << "Valor Invalido" << endl;
        exit(1);
    }

    int qtdRemocoes = n - n1; // Numero total de remoções

    for (int i = 0; i < qtdRemocoes; i++)
    {
```

```
        removeFinal(); //Para cada numero de remoções, retiramos um  
item do final  
    }  
}
```