



2ª Avaliação - Prática (Valor: 2,0)

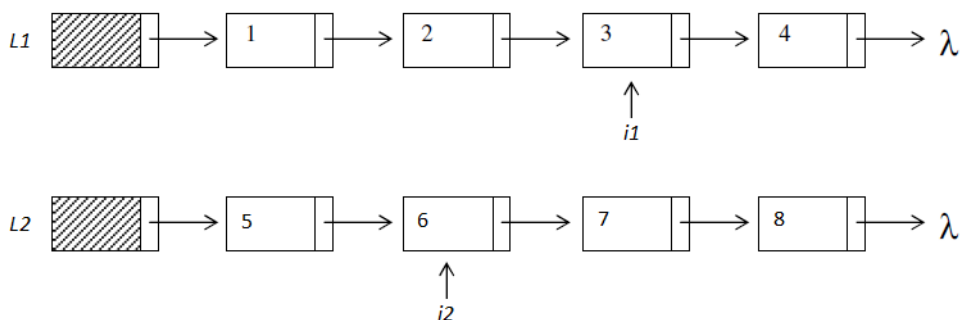
Nome: _____ R.A.: _____

1. Para uma lista encadeada com cabeça, realize as seguintes operações:

a. Escreva uma rotina, `inssub(L1,i1,L2,i2,len)` para inserir os elementos da lista `L2` na lista `L1`. O primeiro elemento da lista `L2` começa na posição `i2` e continuando por `len` elementos na lista `L1`, começando na posição `i1`. Nenhum elemento da lista `L1` deverá ser removido ou substituído. (Valor: 1,4)

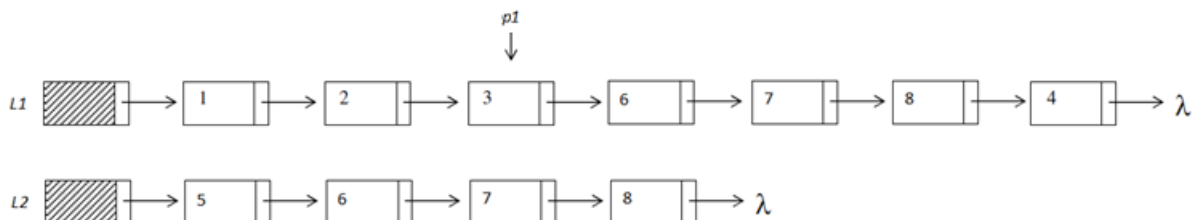
- Se $i1 > \text{length}(L1) + 1$ (onde $\text{length}(L1)$ indicar o número de nós na lista `L1`), ou
- Se $i2 + \text{len} - 1 > \text{length}(L2)$ (onde $\text{length}(L2)$ indicar o número de nós na lista `L2`), ou
- Se $i1 < 1$, ou se $i2 < 1$, imprima uma mensagem de erro.

A lista `L2` deve permanecer inalterada.



*len = número de elementos a serem copiados a partir de i2.
Se len = 3, deverá inserir 3 nós iniciando em i2, neste exemplo, seriam inseridos em L1 os nós de L2 com chave 6, 7 e 8.*

O resultado da execução da função `inssub()`, seria:



Essa função deve obedecer ao protótipo: `void inssub(TNo **L1, i1, TNo *L2, i2, len);`

Importante: a lista `L2` pode ser requisitada pela função `inssub()` no máximo 4 vezes. Sempre que a inserção ocorrer na `L1`, um ponteiro deverá “marcar” a 1ª posição dos



novos elementos inseridos em $L1$. Se podem ocorrer 4 chamadas à função `inssub()`, necessitamos de 4 ponteiros para marcar os pontos de inserções na $L1$.

Sugestão: criar um vetor de ponteiros para controlar os pontos de inserção na $L1$, pois, cada sublista de $L2$ inserida em $L1$ deve ficar “marcada” como ponto de inserção.

b. Crie uma rotina para **desalocar** todos os elementos das listas $L1$ e $L2$. (valor: 0,6)

Observação: no Moodle existe um arquivo chamado `2aaval.c`. Neste arquivo, já estão implementadas as funções de inserção, impressão e a `main()` com menu para as chamadas das funções. Utilize este arquivo como sendo a base para implementar as demais funções solicitadas.