

Universidade Federal do Maranhão  
Centro de Ciências Exatas e Tecnologia  
Departamento de Informática

Curso de Ciência da Computação - Disciplina: Estrutura de Dados  
Segunda Prova.

Pedro Thiago

1. Faça um algoritmo que recebe uma Lista linear Simplesmente Encadeada l1 e retorna uma lista linear simplesmente encadeada l2 com o mesmo comprimento de l1. O k-ésimo nó de l2 possui o valor da soma dos k primeiros nós de l1.

SLList \* SllCumulativeSum( Sllist \*l1, int (\*getvalue)(void \*));

A função getvalue retorna o valor inteiro armazenado no campo data de cada nó.

2. Faça um algoritmo que recebe uma lista linear duplamente encadeada e dois números inteiros i e j. O algoritmo deve trocar a posição do i-ésimo nó da lista pela do j-ésimo nó da lista. Não pode alocar novos nós.

int DllCumulativeSum( Dllist \*l1, int i, int j)

3. Faça um algoritmo que recebe duas listas lineares simplesmente encadeadas l1 e l2 e três valores inteiros i1, i2 e len, e insere os nós da lista l2, começando após o i2-ésimo nó e continuando por len nós, na lista l1, começando antes do i1-ésimo nó. Nenhum elemento da lista l1 deverá ser removido ou substituído. Se  $i1 > \text{length}(l1) + 1$  (onde  $\text{length}(l1)$  indica o número de nós na lista), ou se  $i2 + \text{len} - 1 > \text{length}(l2)$ , ou se  $i1 < 1$ , ou se  $i2 < 1$ , retorne FALSE. Os nós devem ser removidos de l2 e incluídos em l1. Não pode alocar novos nós.

int InSub ( Sllist l1, int i1, int i2, int len)

```
int SllNumElms( Sllist *l1 ) {  
    int out = 0;  
    if (l1 != NULL) {  
        if (l1->first != NULL) {  
            out = l1->first;  
            out++;  
            while (out->next != NULL) {  
                out = out->next;  
                out++;  
            }  
        }  
    }  
    return out;  
}
```