Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnologia
Departamento de Informática
Curso de Ciência da Computação - Disciplina: Estrutura de Dados I

Segunda Prova.

1. Escreva um algoritmo que recebe uma lista circular simplestemente encadeada e conta o numero de vezes que um determinado dado indetificado por *key* se encontra na lista.

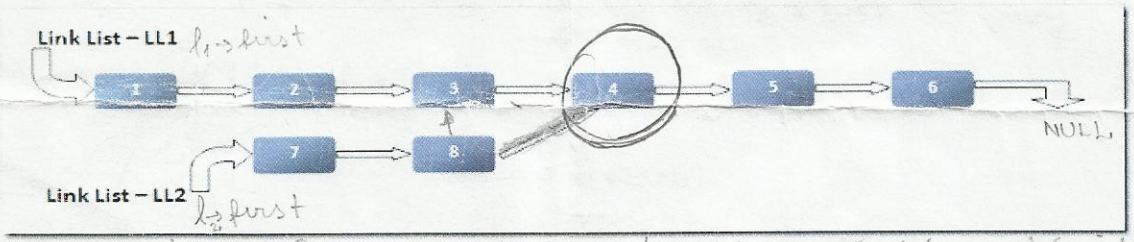
int lcseGetNumerOfOcurrences(SLList *l, void *key, int (*cmp)(void *, void *))

obs: a função cmp retorna TRUE se a chave identifica o dado e FALSE em caso contrário

2. Faca um algoritmo que recebe uma lista circular simplesmente encadeada (L1) e uma lista linear simplesmenete encadeada (L2) e modifica a lista L1 de modo que ela seja uma lista circular simplesmente encadeada com os elementos de L2 adicionados ao final dos seus elementos originais. OBS: não pode alocar novos nós.

void lcseAppendList(SLList *l1, Sllist *L2)

3. Faca um algoritmo que recebe duas listas lineares duplamente encadeadas(LL1 e LL2), como mostrado na figura abaixo, verifica se elas compartilham um nó e encontra o nó que pertence às duas listas removendo-o da lista LL1.




```
int lcseGetNumerOfOcurrences(SLList *l, void *key, int(*cmp)(void*, void*)){
    int cont=0; SLNode *current; int a;
    if(l != NULL){
        if (l->first != NULL){
            current = l->first;
            while(current->next != l->first){
                a = cmp(key, current->data)
                if(a == TRUE){
                    cont = cont + 1;
                }
                current = current->next;
            }
            return cont;
        }
        return FALSE;
    }
```