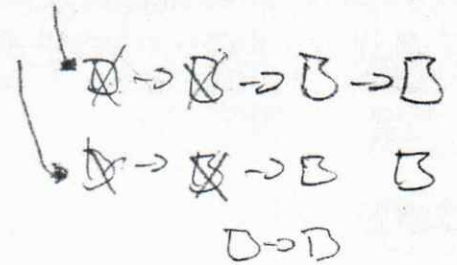


Diuis Glauco Barros Carvalho CPD7204 - 73

UFMA - CCET - Departamento de Informática  
Curso de Ciência da Computação  
Prof. Anselmo Cardoso de Paiva  
Disciplina: Estrutura de Dados I



### Reposição - Segunda Avaliação - 2011.2

1. Faça um algoritmo que recebe duas listas lineares simplesmente encadeadas (L1 e L2) e retorna na lista L3 os elementos que estão ao mesmo tempo em L1 e L2, removendo-os das duas lista anteriores. Pode usar a função de desalocação de nós, mas não pode alocar novos nós. Considere que a lista L3 já está criada e vazia.  
void ComunsDasListas (SLList L1, SLList L2, SLList L3, int (\*cmp)(void \*a, void \*b))  
OBS: a função cmp retorna TRUE se  $a = b$  e FALSE caso contrário.
2. Faça um algoritmo que recebe uma lista linear duplamente encadeada e remove todos os elementos menores que um valor especificado (spec).  
void RemoveMenores (DLList \*L, void \*spec, int (\*menor)(void \*a, void \*b))  
OBS: a função menor retorna TRUE se  $a < b$  e FALSE caso contrário.
3. Faça um algoritmo que recebe uma lista circular simplesmente encadeada e um valor especificado e remove o elemento anterior ao elemento identificado pelo valor especificado.  
void \*RemoveAnterior (SLList \*L, void \*spec, int (\*cmp)(void \*a, void \*b))  
OBS: a função cmp retorna TRUE se  $a = b$  e FALSE caso contrário.

2) void Remove Menores (DLList \*L, void \*spec, int (\*cmp)(void \*a, void \*b))

```
{
    DLLNode *cur;
    IF (L != NULL)
    {
        IF (L->FIRST != NULL)
        {
            cur = L->FIRST;
            WHILE (cur != NULL)
            {
                IF (cmp (spec, cur->DATA) == TRUE)
                {
                    IF (cur == L->FIRST)
                    {
                        L->FIRST = cur->next;
                        cur->next->prev = NULL;
                        free (cur);
                    }
                    ELSE IF (cur->next == NULL)
                }
            }
        }
    }
}
```

```
graph LR
    A --> B
    B --> C
    C --> D
    A -- prev --> NULL
    D -- next --> NULL
    spec --> A
    spec --> B
    spec --> C
    spec --> D
```