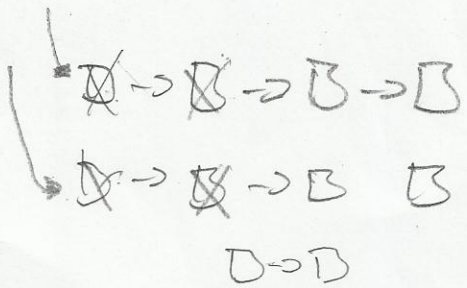
Denis Glauco Barros Carvalho CPD7201-A3



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 e 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*,VOID*))
{
  DLNODE* 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)
```