

EACH - Escola de Artes, Ciências e Humanidades
- Algoritmos e Estruturas de Dados - II - Sistemas de Informação

Prof. Helton Hideraldo Bísaro

4º lista:

1. Escreva em “C”, uma estrutura de dados para uma lista generalizada que pode conter elementos inteiros, do tipo char, do tipo float e do tipo lista.
2. Liste todas as subrotinas necessárias para a implementação de uma lista generalizada em “C”.
3. Implemente em “C” as funções *head*, *tail*, *push* e *addon*, para uma lista generalizada.
4. Escreva uma rotina em “C” chamada *dlt(n)*, que deleta o n-ésimo elemento de uma lista generalizada. Se este elemento for também uma lista, todos os nós acessíveis a partir dela também devem ser liberados.
5. Reescreva as rotinas do exercício 3, para listas duplamente encadeadas.
6. Explique o que é “Garbage Collection” e porque não é uma tarefa trivial.
7. Explique porque não é indicado o uso de recursão ao se fazer “Garbage Collection”.
8. Escreva um código para em “C” para fazer “Garbage Collection” (Escolha um dos algoritmos vistos em sala de aula).
9. Explique o que as rotinas *push(n)* e *pop(n)* devem fazer quando os próprios ponteiros da lista generalizada estão sendo usados para gerenciar a pilha de recursão. Assuma que o ponteiro externo para a lista está na variável *l* e que o topo da pilha é armazenado na variável *top*.
10. Dados *p* e *q*, dois ponteiros para nós de uma lista *l*. Escreva um algoritmo que determina se o nó *q* é acessível a partir do nó *p*.