

Universidade Federal do ABC
MCTA028-15 - Programação Estruturada
2018.Q3

Lista de Exercícios 8

Professores Emílio Francesquini e Carla Negri Lintzmayer

27 de novembro de 2018

1. Escreva uma função que dada uma lista ligada sem cabeça devolva o número de elementos da lista.
2. Escreva uma função que dado um vetor de inteiros devolva uma lista ligada de inteiros cujos elementos são apresentados na mesma ordem do vetor. Faça duas versões: uma iterativa e uma recursiva.
3. Escreva uma função que dada uma lista ligada de inteiros devolva um ponteiro para um vetor de inteiros cujos elementos são apresentados na mesma ordem da lista ligada dada. Faça duas versões: uma iterativa e uma recursiva.
4. Escreva uma função que dado uma lista ligada e um valor v remova da lista todas as células que contêm o valor v . Escreva uma função que remova a k -ésima célula de uma lista encadeada sem cabeça. Faça duas versões: uma iterativa e uma recursiva.
5. Escreva uma função `void inverte(No *cabeca)` que recebe uma lista ligada com cabeça. Sua função deve inverter a ordem dos elementos da lista.
6. Escreva uma versão de `No *selectionsort(No *ini)` que recebe uma lista ligada sem cabeça `ini` e devolve um ponteiro para uma lista ligada ordenada sem cabeça com os nós da lista original. Como você adaptaria este código caso a lista tivesse cabeça?
7. Escreva uma função `void mergesort(No *cabeca)` que recebe uma lista ligada com cabeça e devolve a lista ordenada para isto utilizando o algoritmo de ordenação MergeSort.

8. Escreva uma função `void remove_llcc(No *ini, int k)` que recebe um ponteiro para uma lista ligada circular com cabeça `ini` e remove o primeiro nó com chave `k`.
9. Escreva uma função `void remove_ldlcc(NoD *p)` que recebe um ponteiro para um nó `p` de uma lista duplamente ligada circular com nó cabeça e o remove da lista. Naturalmente, suponha que `p` não é o nó cabeça da lista.
10. Escreva uma função `insere_lcco(No *ini, int k)` que recebe uma lista ligada circular com cabeça ordenada `ini` e insere um novo nó com chave `k` na posição correta.
11. Problema de Josephus:
 - Um grupo de N pessoas precisa eleger um líder.
 - Decidiu-se usar a seguinte ideia para eleger um líder: forma-se um círculo com as N pessoas e escolhe-se um inteiro k . Começamos com uma pessoa qualquer e percorremos o círculo em sentido horário, eliminando cada k -ésima pessoa. A última pessoa que restar será o líder. Veja o verbete sobre Josephus na Wikipedia.

Coloque os números $1, 2, \dots, N$ em um círculo nesta ordem e, começando em 1, aplique o algoritmo acima com um valor k . Determine o último número, denotado $J(N, k)$. Escreva uma função `int Josephus(int N, int k)` que calcula $J(N, k)$.