



Universidade Federal de Mato Grosso
Instituto de Computação

Disciplina: Algoritmos II

Professora: Vanessa de Oliveira Campos

LISTA DE EXERCÍCIOS

ALOCAÇÃO DINÂMICA DE MEMÓRIA

1. Faça o algoritmo de uma função lógica que inclua um elemento no final de uma lista.
2. Faça o algoritmo de um procedimento que exclua de uma lista de inteiros, todos os elementos que tenham o mesmo valor passado como parâmetro.
3. Faça o algoritmo de um procedimento que esvazie uma lista.
4. Faça o algoritmo de uma função que retorne a quantidade de elementos da lista.
5. Faça o algoritmo de uma função lógica que inclua um elemento na posição i de uma lista.
6. Faça o algoritmo de uma função que, a partir de duas listas simplesmente encadeadas, cria uma terceira lista que seja a união sem repetição dos elementos das duas primeiras.
7. Faça o algoritmo de uma função inteira que retorne a posição de um determinado elemento x real aparece na lista encadeada simples. Retorne -1 em caso de erro.
8. Faça o algoritmo de uma função real [iterativa/recursiva] para retornar a média aritmética simples dos valores inteiros de uma lista encadeada simples. Retorne 0 no caso da lista estar vazia.
9. Faça o algoritmo de uma função lógica para dizer se um determinado elemento x aparece junto de um elemento y , nesta ordem, na lista encadeada simples de reais.
10. Faça o algoritmo de uma função lógica [iterativa/recursiva] que diz se todos os elementos de uma lista simplesmente encadeada ordenada de reais encontram-se contidos em outra lista simplesmente encadeada também ordenada.
11. Faça o algoritmo de uma função lógica [iterativa/recursiva] para remover um elemento da posição p de uma lista [simplesmente/duplamente] encadeada de inteiros, retornando VERDADEIRO se o elemento foi removido com sucesso.
12. Faça o algoritmo de uma função lógica para inserir um elemento no início de uma lista duplamente encadeada.

13. Faça o algoritmo de uma função lógica para inserir um elemento inteiro na posição ordenada de uma lista duplamente encadeada que está ordenada.
14. Faça o algoritmo de uma função lógica [iterativa/recursiva] para remover todos os elementos que aparecem em uma lista [simplesmente/duplamente] encadeada a de uma outra lista b , o que equivale à operação $b - a$ em conjuntos, retornando VERDADEIRO se pelo menos um elemento foi removido com sucesso.
15. Faça o algoritmo de uma função lógica [iterativa/recursiva] para remover todos os elementos que aparecem após um elemento a e antes de um elemento b em uma lista simplesmente encadeada, retornando VERDADEIRO se pelo menos um elemento foi removido com sucesso.
16. Faça o algoritmo de uma função inteira [iterativa/recursiva] para remover todos os elementos duplicados que aparecem em uma lista simplesmente encadeada de reais ordenada de forma ascendente, retornando a quantidade de elementos (nós) efetivamente retirados.
17. Faça o algoritmo de um procedimento que faça com que o último elemento de uma lista simplesmente encadeada de reais, recebida por parâmetro, vá para a primeira posição da mesma.
18. Faça o algoritmo de um procedimento que faça com que o primeiro elemento de uma lista [simplesmente/duplamente] encadeada de reais vá para a última posição da mesma, modificando-a.
19. Faça o algoritmo de um procedimento [iterativo/recursivo] para inverter os elementos de uma lista simplesmente encadeada de caracteres passada como parâmetro. Por exemplo, se a lista entregue é $\{a,b,c,d,e\}$ então a lista devolvida é $\{e,d,c,b,a\}$. Faça as devidas definições de tipo previamente à declaração da função.
20. Faça o algoritmo de uma função lógica para juntar dois elementos consecutivos de uma lista simplesmente encadeada de reais, das posições p e $p+1$, somando seus valores, em um único nó. Caso alguma das posições não exista, nada deve ser feito. A função deve retornar VERDADEIRO apenas no caso da junção ter sido feita.
21. Faça o algoritmo de uma função ponteiro de duplicação de uma lista [simplesmente/duplamente] encadeada de reais, passada por parâmetro, retornando o ponteiro para a lista-cópia, ou NULO em caso de erro.
22. Faça um algoritmo para dividir uma lista [simplesmente/duplamente] encadeada em duas metades, formando duas listas encadeadas.
23. Dada uma lista simplesmente encadeada de inteiros, faça um algoritmo para criar outras duas que contenham, respectivamente, somente os números pares e somente os números ímpares, mantendo a ordem relativa original dos elementos nas novas listas criadas.
24. Faça o algoritmo de uma função inteira para remover todos os elementos duplicados que aparecem em uma lista simplesmente encadeada de reais ordenada de forma ascendente, retornando a quantidade de elementos (nós) efetivamente retirados.