

## Lista de Exercícios – Listas Encadeadas – FATEC SJC – ED 2011

(pode ser feita em grupo, com a condição que cada um entenda o que foi feito)

1. REMOVE DUPLICADOS. Escreva um código que remova elementos duplicados em uma lista encadeada não ordenada. Suponha a lista com cabeça.
2. SOMA. Sejam dois números armazenados em listas ligadas com cabeça, onde cada conteúdo é um dígito. Os dígitos estão armazenados na ordem reversa. Escreva um código que some os dois números e retorne uma lista ligada com o resultado. Exemplo: 3->1->5 + 5->9->2 = 8->0->8
3. BUSCA LISTA SEM CABEÇA. Escreva uma versão da função Busca para listas sem cabeça.
4. **MÍNIMO**. Escreva uma função que encontre uma célula de conteúdo mínimo. Faça duas versões: uma iterativa e outra recursiva.
5. LISTA CRESCENTE. Uma lista é *crescente* se o conteúdo de cada célula não é maior que o conteúdo da célula seguinte. Escreva uma função que faça uma busca em uma lista crescente. Faça versões para listas com e sem cabeça. Faça uma versão recursiva e outra iterativa.
6. REMOVA LISTA SEM CABEÇA. Escreva uma função que remova uma determinada célula de uma lista encadeada sem cabeça. (Será preciso tomar algumas decisões de projeto antes de começar a programar).
7. INSIRA ANTERIOR. Escreva uma função que insira uma nova célula entre a célula cujo endereço é *p* e a *anterior*.
8. INSIRA LISTA SEM CABEÇA. Escreva uma função que insira uma nova célula numa dada posição de uma lista encadeada sem cabeça. (Será preciso tomar algumas decisões de projeto antes de começar a programar).
9. **VETOR PARA LISTA**. Escreva uma função que copie um vetor para uma lista encadeada.
10. LISTA PARA VETOR. Escreva uma função que copie uma lista encadeada para um vetor.
11. CÓPIA. Escreva uma função que faça uma cópia de uma lista dada.
12. COMPARAÇÃO. Escreva uma função que decida se duas listas dadas têm o mesmo conteúdo.
13. **CONCATENAÇÃO**. Escreva uma função que concatene duas listas encadeadas (isto é, “amarre” a segunda no fim da primeira).
14. CONTAGEM. Escreva uma função que conte o número de células de uma lista encadeada.
15. PONTO MÉDIO. Escreva uma função que receba uma lista encadeada e devolva o endereço de uma célula que esteja o mais próximo possível do ponto médio de uma lista. Faça isso sem calcular explicitamente o número *n* de células da lista e o quociente  $n/2$ .
16. CONTAGEM E REMOÇÃO. Escreva uma função que remova a *k*-ésima célula de uma lista encadeada.
17. CONTAGEM E INSERÇÃO. Escreva uma função que insira uma nova célula com conteúdo *x* entre a *k*-ésima e a (*k*+1) ésima células de uma lista encadeada.
18. **LIBERAÇÃO**. Escreva uma função que aplique a função *free* a todas as células de uma lista encadeada. Estamos supondo, é claro, que cada célula da lista foi originalmente alocada por *malloc*.
19. **INVERSÃO**. Escreva uma função que inverta a ordem das células de uma lista encadeada (a primeira passa a ser a última, a segunda passa a ser a penúltima etc.). Faça isso sem criar novas células; apenas altera os ponteiros.
20. **PROBLEMA DE JOSEPHUS**. Imagine *n* pessoas dispostas em círculo. Suponha que as pessoas estão numeradas de 1 a *n* em sentido horário. Começando com a pessoa de número 1, percorra o círculo no sentido horário e elimine a *m*-ésima pessoa enquanto o círculo tiver duas ou mais pessoas. Veja *Josephus Problem* na Wikipedia. Qual o número do sobrevivente para *n*=50 e *m*=3? Escreva e teste uma função que resolva o problema.
21. Leia o verbete *Linked list* na Wikipedia.