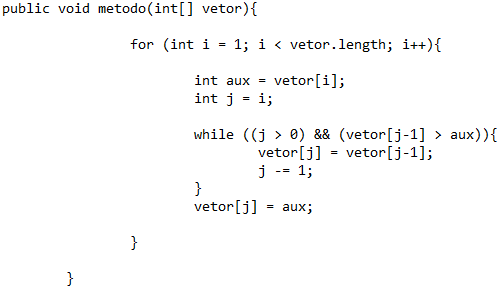
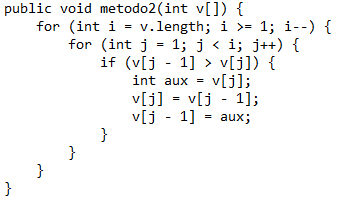
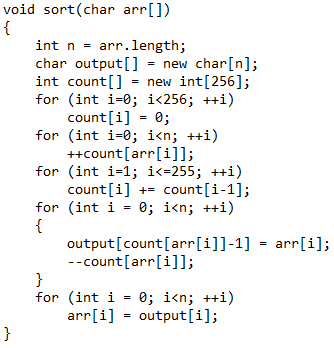
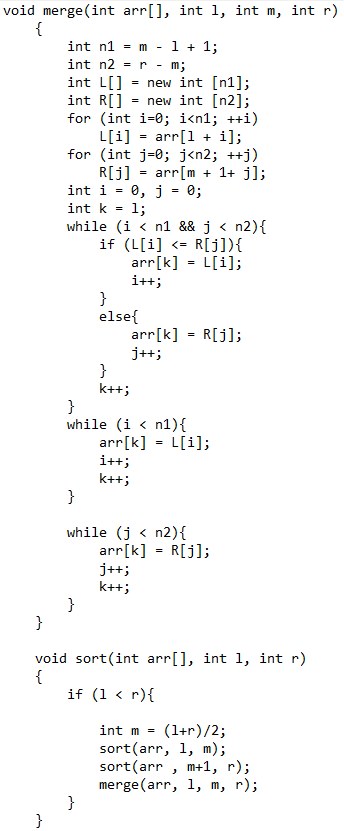
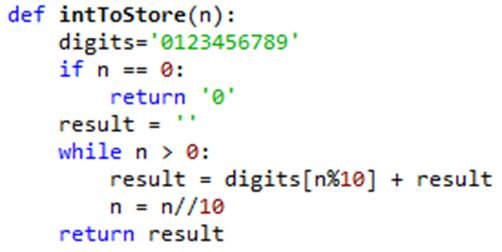
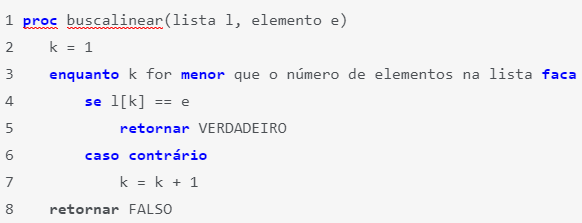
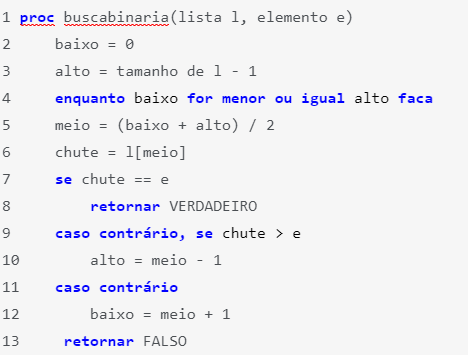
**EXERCÍCIO - Análise de Algoritmos 1**

1. Por que realizar testes para computar o tempo de execução de um algoritmo é uma abordagem complexa e pouco confiável?
2. Para contabilizar a complexidade de tempo de um algoritmo, pode-se contabilizar os custos de cada instrução e soma-los. Nesse contexto, qual a utilidade da análise assintótica?
3. Para cada uma das seguintes situações, indique se f = O(g), ou f = Ω(g) ou ambos (caso em que f = Θ(g)).
   1. f(n) = n – 100; g(n) = n – 200
   2. f(n) = n2; g(n) = n3
   3. f(n) = nlogn; g(n) = 10nlog10n
   4. f(n) = n1/2; g(n) = n3/2
   5. f(n) = 2n ; g(n) = 2n+1
4. Indique o limite assintótico superior do tempo de execução para os trechos de código abaixo:
   1. 
   2. 
   3. 
   4. 
   5. 
   6. 
   7. 
   8. 
   9. 
5. Ordene os métodos/funções da questão 2 de mais rápido para mais lento considerando o limite superior assintótico do tempo de execução. (Exemplo de resposta: “abcfegd”)
6. O objetivo de utilizar a análise assintótica para realizar a estimativa do tempo de execução de um algoritmo é que a mesma ignora os termos constantes e de ordem inferior, simplificando a análise e tendo resultados aceitáveis para o pior caso. Prove que essa afirmação é verdadeira.
7. Considere uma alteração no Merge Sort para que, ao invés de dividir o array ao meio a cada passo do algoritmo, a divisão seja em terços; ordena-se cada terço, e finalmente, combina-se todos utilizando uma subrotina de merge. Qual é o tempo de execução (assintótico) deste algoritmo? (Dica: note que cada passo de merge pode ser implementando em O(n).)