

Programação de Computadores I

2ª lista de exercícios

Para os problemas a seguir, a não ser que o enunciado instrua de forma diferente, faça o programa em Java, assim como os testes necessários para assegurar que a solução está correta.

1. Faça uma função que calcule a média de um aluno de acordo com o critério definido neste curso. Além disso, faça uma segunda função que informe o status do aluno de acordo com a tabela a seguir:

Nota acima de 6 → “Aprovado”

Nota entre 4 e 6 → Conceito “Verificação Suplementar”

Nota abaixo de 4 → Conceito “Reprovado”

2. Leia do usuário o tempo em segundos e escreva em horas, minutos e segundos. Utilize cinco métodos, para a leitura e escrita de dados e para obtenção de horas, minutos e segundos a partir do tempo em segundos.
3. Leia uma matriz 3 x 3 que representa um tabuleiro de jogo da velha e indique qual posição deveria ser jogada para ganhar o jogo (se possível) ou ao menos para evitar uma derrota.
4. Ordene um vetor de 100 números inteiros gerados aleatoriamente. Esse programa deve implementar o algoritmo selection sort. Esse algoritmo faz uso de uma função para selecionar o menor elemento a partir de cada posição do vetor e inseri-lo nesta posição.
5. Leia um número decimal (até 3 dígitos) e escreva o seu equivalente em numeração romana. Utilize funções para obter cada dígito do número decimal e para a transformação de numeração decimal para romana (Dica1: 1 = I, 5 = V, 10 = X, 50 = L, 100 = C, 500 = D, 1.000 = M; Dica2: utilize um vetor guardando a tradução para cada um dos dígitos).
6. Escreva um número por extenso aceitando números de até 9 dígitos, usando métodos para as traduções e vetores de Strings que guardam cada tradução (ex.: unidades = { “zero”, “um”, “dois”, ..., “nove” }).
7. Crie dois vetores de 50 posições com valores inteiros aleatórios, ordene cada vetor individualmente, e combine os dois vetores gerando um novo vetor de 100 posições, de forma que esse novo vetor já seja criado ordenado. Utilize subprogramação sempre que possível.
8. Ordene um vetor de números inteiros com 100 posições geradas aleatoriamente utilizando o algoritmo QuickSort, sem criar vetores extras. Dica: separe o primeiro elemento do vetor (pivô), divida o vetor em dois: vetor de menores e vetor de maiores que o pivô, ordene recursivamente cada vetor (menores e maiores) e concatene o vetor de menores ordenado com o pivô e com o vetor de maiores ordenado.

9. Calcule a raiz quadrada aproximada de um número inteiro informado pelo usuário, respeitando o erro máximo também informado pelo usuário. Não utilize funções predefinidas.
10. A partir de dois textos informados, detecte a maior sequência comum a esses textos. Por exemplo, para "CASAMENTO" e "CANTORIA", a maior sequência comum é "CANTO". Dica: Use recursão!