DISCIPLINA: Programação Avançada

4 º PERIDO – SISTEMAS DE INFORMAÇÃO - NASSAU

PROF. MSC. DOUGLAS SANTIAGO KRIDI

OBS.: Entrega *solo* ou em *duplas*

Vetores e Matrizes

Vetores

1. Escreva um programa que leia 10 números inteiros e os armazene em um vetor. Imprima o vetor, o maior elemento e a posição que ele se encontra.

2. Faça um programa que preencha um vetor com 10 números reais, calcule e mostre a quantidade de números negativos e a soma dos números positivos desse vetor.

3. Leia um vetor com 10 números inteiros. Escreva os elementos do vetor eliminando elementos repetidos.

4. Faça um vetor de tamanho 50 preenchido com o seguinte valor: *(i + 5 * i)%(i + 1)*, sendo *i* a posição do elemento no vetor. Em seguida imprima o vetor na tela.

5. Faça um programa que leia dois vetores de 10 elementos. Crie um vetor que seja a intersecção entre os 2 vetores anteriores, ou seja, que contém apenas os números que estão em ambos os vetores. Não deve conter números repetidos.

6. Faça um programa que leia dois vetores de 10 elementos. Crie um vetor que seja a união entre os 2 vetores anteriores, ou seja, que contém os números dos dois vetores. Não deve conter números repetidos.

7. Elabore uma classe chamada *MediaAluno* que contenha um atributo do tipo vetor de inteiros com o nome de *notas*. Essa classe deve ter:
    a) Um método para adicionar as notas nesse vetor (os valores que podem ser adicionados no vetor são os inteiros entre 0 e 10, caso contrário imprime uma mensagem de erro e não adiciona)
    b) Outro método que calcule a média de um aluno e imprima essa média.
    c) Considere que o vetor deve receber 8 notas.

Matrizes

8. Crie um método que recebe uma matriz bidimensional *double* e retorna a quantidade de linhas da matriz.

9. Crie uma matriz M 3x3. Após a criação, imprima a matriz criada e encontre a quantidade de números pares e a quantidade de números ímpares.

10. Faça um programa para gerar automaticamente números entre 0 e 99 de uma cartela de bingo. Sabendo que cada cartela deverá conter 5 linhas de 5 números, gere estes dados de modo a não ter números repetidos dentro das cartelas. O programa deve exibir na tela a cartela gerada.

Questão de Revisão

11. Desenvolva uma aplicação que simule um jogo tradicional envolvendo os elementos: Papel, Pedra e Tesoura.
    a) Considere o seguinte:
        - A aplicação deverá ter uma classe ***Elemento***, a qual será um supertipo dos elementos envolvidos no jogo (*papel*, *pedra* e *tesoura*).
        - Deverá ser implementada uma classe ***Jogo*** que terá um método ***Jogar*** que receberá dois elementos como parâmetro, sem saber de que tipo são, e dependo do confronto deverá exibir uma *String* informando: "*Papel ganhou, pois envolve Pedra*.", "*Tesoura ganhou, pois corta Papel*.", "*Pedra ganhou, pois quebra Tesoura*".
        - Caso os elementos confrontados sejam do mesmo tipo o jogo terminará empatado.

    b) Implemente uma classe aplicativo que ilustre a execução do jogo considerando o seguinte:
        - Criar um objeto de cada tipo dos elementos envolvidos no jogo.
        - Criar um objeto do tipo jogo
        - Ativar o método jogar, do objeto jogo criado anteriormente, passando como parâmetro as combinações possíveis entre os elementos envolvidos no jogo, por exemplo, *jogar(papel, tesoura)*.
        - Execute e teste todas as combinações possíveis entre os elementos:
            - Papel x Pedra, Papel x Tesoura, Papel x Papel
            - Pedra x Papel, Pedra x Tesoura, Pedra x Pedra
            - Tesoura x Pedra, Tesoura x Papel, Tesoura x Tesoura.