

Lista de Exercícios II – Problemas envolvendo vetores

[1] Crie um programa que leia um vetor de 10 posições inteiras. Inverta o vetor, alterando os valores de posição. Mostre o novo vetor.

[2] Elabore um programa que leia um vetor de 10 posições inteiras. Depois, solicite para o usuário um número que ele gostaria de pesquisar neste vetor, caso o número exista no vetor, mostre em qual(is) posição(ões) ele foi encontrado e quantas ocorrências foram detectadas.

[3] Construa um programa que sugira uma aposta de Mega-Sena (popularmente apelidada de surpresinha), ou seja, um algoritmo que gera e mostra um conjunto de 6 números aleatórios entre [1, 60] sem repetição.

[4] Desenvolva um programa que leia 10 números inteiros e armazene-os em um vetor chamado **vLido**. Depois, crie dois outros vetores: **vPares**, contendo somente os números pares de **vLido**, e **vImpares** contendo somente os números ímpares de **vLido**. Os vetores **vPares** e **vLido** não deverão conter zeros. Mostre então os três vetores.

[5] Escreva um programa que leia um vetor de 10 posições reais, chamado **vet_In**, calcule a média aritmética do vetor. Num outro vetor chamado **vet_Des** você deverá armazenar as diferenças absolutas do valor original para a média calculada. Fazendo então o cálculo da média deste novo vetor, que representa a média aritmética dos valores absolutos dos desvios em relação à média, ou seja, o desvio médio do conjunto inicial. Mostre então ao usuário qual foi o desvio médio do conjunto que ele digitou.

Observação: o desvio médio absoluto (DMA) é uma medida de dispersão que leva em conta todos os valores em um conjunto de dados. A estratégia apresentada acima resume algoritmicamente como ele é calculado através da fórmula:

$$DMA = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|$$

[6] Escreva um programa que leia um vetor de 10 posições com valores inteiros, aceitando apenas números entre 1 e 5 (inclusive). Depois, em um vetor chamada **vFreq** calcule a distribuição de frequência das ocorrências dos valores possíveis para o vetor. Mostre então esta distribuição calculada em **vFreq** e imprima um histograma horizontal (modo texto) representando esta distribuição.

[7] Escreva um programa que preencha um vetor de números inteiros de 10 posições aceitando apenas valores positivos. Modifique então o vetor de forma que, tenhamos primeiro todos os números pares, depois, os números ímpares. Mostre o vetor antes e depois da modificação.

[8] Escreva um algoritmo que leia o número de linhas (**n**) que o usuário deseja e então mostre o Triângulo de Pascal com n linhas. Exemplo: para n igual a 6, temos:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

(Para maiores informações sobre o Triângulo de Pascal: https://pt.wikipedia.org/wiki/Triângulo_de_Pascal)

[9] Elabore um programa que preencha um vetor de 100 posições inteiras com números aleatórios no intervalo [100, 999] e que então coloque este vetor em ordem crescente. Utilize como estratégia de ordenação a comparação de pares de elementos adjacentes, permutando-os quando estiverem fora de ordem, até que todos estejam ordenados.

Observação: este método de ordenação é conhecido como “*Buble Sort*”, método da bolha, ou ordenação por trocas.

[10] Desenvolva um programa que leia um vetor de 10 posições reais e o coloque em ordem crescente, utilizando a seguinte estratégia de ordenação:

- selecione o elemento do vetor de 10 posições que apresenta o menor valor;
- troque este elemento pelo primeiro;
- repita estas operações, envolvendo agora apenas os 9 elementos restantes (trocando o de menor valor com a segunda posição), depois os 8 elementos (trocando o de menor valor com a terceira posição), depois os 7, 6 e assim por diante, até restar um único elemento, o maior deles.

Observação: este método de ordenação é conhecido como “*Seleção Direta*”.