

CI208 – Programação de Computadores
2º Período Especial (2020-2021)

Exercícios #08
Vetores e Funções

Esta lista de exercícios será trabalhada durante as aulas online.

Estes exercícios não valem nota ou frequência e serão usados para esclarecer e consolidar os conceitos abordados nos diversos materiais da **Sala de Estudos**.

Você também pode explorar os exercícios que estão no **material complementar**, também na **Sala de Estudos**.

Nos exercícios abaixo, os nomes entre parênteses no início do enunciado são sugestões para os nomes dos programas solicitados.

1. **(escalar)** Crie a função `escalar` que receba como parâmetros dois vetores A e B de N posições com números reais (N definido via `#define`) e devolva o valor do produto escalar entre os dois vetores. O produto escalar entre dois vetores é dado por:

$$x_0y_0 + x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4 + x_5y_5 + \dots$$

Faça também o programa principal que recebe do usuário dois vetores A e B de N posições e usando a função `escalar` calcule e mostre na tela o valor do produto escalar entre A e B . Defina também uma função apropriada para a leitura de vetores.

2. **(pertence)** Faça um programa que preencha um vetor A de até MAX posições com números inteiros positivos (MAX definido via `#define`).

Em seguida o programa deve ler números do teclado e para cada número lido indicar ao usuário se ele está ou não no vetor A . Nesse ponto, a digitação de um número negativo deve causar o término do programa.

Para verificar se um valor está em um vetor, defina a função `pertence` que recebe um vetor de inteiros, sua quantidade de elementos e um valor inteiro v . A função deve retornar **1** (um) se o valor v pertence ao vetor, **0** (zero) caso contrário.

Para a leitura do vetor, crie e use no programa a função **lerVetor** que efetua a leitura de valores inteiros em um vetor de MAX posições até que seja digitado um valor negativo, que não deve ser armazenado no vetor. A função deve devolver o vetor preenchido e a quantidade de elementos armazenados no vetor.

3. **(intersecao)** Faça um programa que preencha dois vetores A e B de N posições com números inteiros (N definido via `#define`). Estes vetores representam conjuntos e seus elementos estão em uma ordem qualquer.
O programa deve calcular e imprimir um vetor C tal que $C = A \cap B$, ou seja, C deve conter a interseção entre os conjuntos A e B .
Devem ser criadas e usadas no programa a função de leitura de vetores (veja exercícios acima), uma função para a exibição dos valores de um vetor na tela, uma função verificar se um valor pertence a um vetor, e um função que cria um vetor que é a intersecção de dois outros vetores
4. **(ordenalInter)** Altere o programa anterior para exibir ao final o conteúdo dos três vetores A , B e C ordenados.