

## Aula 21: Estruturas Complexas (vetores)

**Exercício 1.** Escreva um programa em C que: inicializa um vetor de tamanho 5 com valores quaisquer digitados pelo usuário; imprime o vetor criado; imprime o vetor criado multiplicado por 2. Para preencher um vetor siga o exemplo, onde TAM é o tamanho do vetor:

```
for(i = 0; i < TAM; i++) {  
    printf("Digite vetor[%d]: ", i);  
    scanf("%d", &vetor[i]);  
}
```

**Exercício 2.** Escreva um programa em C que declare 3 vetores quaisquer de tamanho 10 e imprime a média entre os 3 vetores posição por posição.

**Exercício 3.** Escreva um programa em C que declara um vetor com tamanho 8, inicializado (na declaração) com valores inteiros quaisquer. O programa deve imprimir todos os valores do vetor incrementados em 5 unidades cada.

**Exercício 4.** Escreva um programa em C que declara um vetor de inteiros de tamanho 20 cujo conteúdo segue a regra:  $vetor[i] = i + 3$ . Como saída o programa deve imprimir todo o vetor gerado em uma 1ª linha, e na 2ª linha deve imprimir apenas os valores múltiplos de 3 pertencentes ao vetor.

**Exercício 5.** Escreva um programa em C que preencha dois vetores de tamanho 4, cujos valores são digitados pelo usuário. Os dois vetores devem ser comparados (posição por posição), e o menor dos valores é impresso (para cada posição).

**Exercício 6.** Escreva um programa em C que descobre o maior dentre 10 valores inteiros quaisquer. Ao invés de declarar 10 variáveis int, declare um vetor com 10 posições e utilize o mesmo para descobrir o maior dos valores.

**Exercício 7.** Escreva um programa em C que leia diversos números do usuário, e preencha um vetor com estes números. Inicialmente, o programa aloca um vetor de tamanho N, o qual será preenchido por valores informados pelo usuário. Se o usuário digitar um valor  $\geq 0$ , este valor deve ser alocado no vetor. Se o usuário digitar um valor  $< 0$ , este valor não deve ser alocado no vetor e a entrada de dados é encerrada, ou seja, um valor negativo é interpretado como um finalizador para a entrada de dados. O processo de leitura também deve parar se acabar o espaço no vetor (isto é, todas as N posições tiverem sido preenchidas). Ao final deste processo, o programa deve escrever duas informações: o número de elementos inseridos no vetor, e os elementos do vetor em si.

**Exercício 8.** Escreva um programa em C que, a partir de dois vetores de mesmo tamanho 10, determine se ambos vetores são idênticos (mesmos elementos nas mesmas posições) ou não. Os vetores podem ter sido preenchidos na inicialização ou então preenchidos pelo usuário (à sua escolha). O programa deve escrever “vetores idênticos” ou “vetores diferentes”.

**Exercício 9.** Escreva um programa em C que, a partir de dois vetores V1 de tamanho M e V2 de tamanho N determine se todos os números que aparecem em V1 também aparecem em V2 (não necessariamente na mesma posição). Os vetores podem ter sido preenchidos na inicialização ou então preenchidos pelo usuário (à sua escolha). O programa deve escrever “V1 é subconjunto de V2” ou “V1 não é subconjunto de V2”.  
Nota: desconsidere a questão de V1 ou V2 possuírem elementos repetidos ou não.

**Exercício 10.** Escreva um programa em C que leia um número inteiro positivo  $X$  digitado pelo usuário, onde  $X \leq 10000$ , e imprima todos os números primos menores ou igual ao número lido. Dica: utilize um vetor para simular o Crivo de Eratóstenes ([http://pt.wikipedia.org/wiki/Crivo\\_de\\_Eratóstenes](http://pt.wikipedia.org/wiki/Crivo_de_Eratóstenes)).

**Exercício 11.** Escreva um programa em C que leia 10 números inteiros do usuário e, após a leitura, imprima os números separadamente como números pares e números ímpares.

**Exercício 12.** Escreva um programa em C que declara (e inicializa da maneira que você achar mais conveniente) um vetor de tamanho N. O programa deve ler do usuário um valor X e procurá-lo no vetor. Caso o valor seja encontrado no vetor deve-se informar a posição onde o valor está armazenado.

**Exercício 13.** Desenvolva um programa em C que declara dois vetores (A e B) de tamanho 6 (inicializados na declaração). Um terceiro vetor (C), também de 6 posições deverá ser declarado, e seu conteúdo deverá ser a soma dos vetores A e B.

**Exercício 14.** Desenvolva um programa em C que percorre um vetor de tamanho 10 com valores inteiros quaisquer (inicializados na declaração); e imprime como resultado o número de vezes que os números 3 ou 1 são encontrados no vetor.