



Vetores

Um **vetor** (unidimensional) pode ser definido abstratamente como um conjunto finito e ordenado de elementos homogêneos:

- **finito**: existe um número específico de elementos no vetor.
- **ordenado**: os elementos são organizados de forma a que exista um elemento zero, um primeiro elemento, um segundo, um terceiro e assim por diante.
- **homogêneo**: todos os elementos no vetor precisam ser de um mesmo tipo.

Declaração de um vetor para armazenar 100 números inteiros em linguagem C:

```
int a[100];
```

A linguagem C permite que vetores sejam preenchidos na declaração. Assim, pode-se definir um vetor para armazenar o número de dias em cada mês:

```
int numdias[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

Operações básicas

As duas operações básicas de um vetor são a **extração** e o **armazenamento**. Uma operação de armazenamento aceita um vetor, um índice e um valor a ser armazenado. Da mesma forma, a extração aceita um vetor, um índice, porém retorna o valor correspondente ao índice no vetor.

O exemplo abaixo mostra um segmento de um programa para **declarar, armazenar e extrair** dados de um vetor:

```
int i, a[100];           // declaração do vetor a e da variável i

for (i=0;i<100;i++)      // armazenamento de dados no vetor
    a[i]=0;

for (i=0;i<100;i++)      // extração (leitura) dos dados do vetor
    printf("\n vetor[%d] = %d ", i, a[i]);
```

Vetor como parâmetro

Todos os parâmetros de uma função em linguagem C necessitam ser declarados dentro da função. Entretanto, a faixa de parâmetros de um vetor unidimensional só é especificada no programa chamador. Isso ocorre porque não é alocado novo espaço de armazenamento para um parâmetro vetor.

Exemplo:

```
float media (int a[], int tam) {
    int i;
    float soma;

    soma = 0;
    for (i=0;i<tam;i++)
        soma += a[i];
    return (soma/tam);
}
```

Programa Principal:

```
...
int vetor[MAX];
float m;
...
m = media (vetor,MAX);
...
```

Exercícios em linguagem C:

- 1) Declare um vetor de **inteiros com 100 posições alocado estaticamente** e preencha-o com **n** valores aleatórios . Em seguida faça funções para:
 - a) Imprimir os elementos contidos no vetor, **k elementos por linha**
 - b) Calcular quantos dos elementos do vetor são pares e retornar o resultado
 - c) Calcular quantos dos elementos do vetor são múltiplos de 3 e retornar o resultado
 - d) Calcular a soma dos elementos de vetor e retornar o resultado
 - e) Calcular a média dos elementos do vetor e retornar o resultado
 - f) Criar um novo vetor contendo apenas os números ímpares e imprimir o novo vetor (usar a função de impressão do item (a)).

Obs.: Para a geração de números aleatórios utilize as funções *rand()* e *srand()* da biblioteca *stdlib.h*

- 2) Dados dois vetores de tamanhos diferentes, faça um programa que imprima os elementos comuns aos dois vetores.