



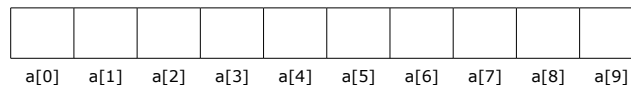
1) Vetores

Um vetor é um conjunto finito de elementos homogêneos. O conjunto é organizado de forma que exista um elemento zero, um primeiro elemento, um segundo e assim por diante. A declaração de um vetor unidimensional é feita da mesma forma que uma variável simples, e segue a sintaxe:

```
tipo nome_vetor [nro_de_elementos];
```

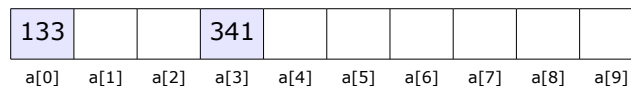
Assim, a declaração em C especifica um vetor de 10 inteiros:

```
int a[10];
```



Em C, os índices de um vetor com n elementos variam de 0 a $n-1$, sendo que o índice do primeiro elemento é sempre 0 (zero). Os índices podem ser utilizados para ler e armazenar dados no vetor:

```
int a[10];  
a[0] = 133; // armazena o valor 133 na primeira posição do vetor a.  
a[3] = 341; // armazena o valor 341 na quarta posição do vetor a.
```



As duas operações básicas para acesso aos elementos de um vetor são a **extração** e o **armazenamento**, e ambas utilizam um índice para localizar uma posição no vetor. O trecho de programa abaixo exemplifica as operações de declaração e atribuição do valor 0 à todas as posições do vetor, e em seguida imprime os valores armazenados:

```
#include <stdio.h>  
  
int main(){  
    int i, a[100];  
  
    for (i=0;i<100;i++); // preenche o vetor com zeros  
        a[i]= 0;  
    for (i=0;i<100;i++) // impressão do vetor  
        printf("%d ",a[i]);  
}
```

Um vetor unidimensional é normalmente utilizado para manter uma grande quantidade de itens na memória, e para referenciar todos os itens de maneira uniforme. Uma declaração

```
int vetor[100];
```

indica que são reservadas 100 posições sucessivas de memória, cada uma com tamanho definido para armazenar um número inteiro. O endereço da primeira dessas posições é chamado *endereço_base*. Assim, uma referência à posição $a[i]$ devolve o elemento em *endereço_base + (tam * i)*.

O trecho de programa abaixo exemplifica a geração aleatória de valores entre 0 e 10 a serem armazenados em um vetor de números reais, de tamanho 100, e o cálculo da média dos valores gerados:

```
#include <stdio.h>
#include <time.h>
int main(){
    float a[100], soma=0, media;
    int i;
    srand (time(NULL));
    for (i=0;i<100;i++){
        a[i]=rand()%11;    // Gera número de 0 a 10 e armazena no vetor a
    }
    for (i=0;i<100;i++){
        soma += a[i];
    }
    media = soma/tam;
    printf("Media = %f", media(a,100));
}
```

Pode-se declarar um vetor sem indicar a quantidade de elementos a ser reservada, desde que os elementos a serem armazenados estejam definidos na sua carga inicial. Neste caso, o compilador calcula o número de elementos que o vetor terá.

```
int a[] = {13,24,35,46};
```

13	24	35	46						
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]

Não se pode declarar vetores sem dimensão ou sem a dimensão calculada a partir da carga automática. Se não for informada a quantidade pretendida, como poderá o compilador definir o espaço necessário na memória?

Exercícios em linguagem C:

1) Faça um programa para gerar um vetor de inteiros randômicos com tamanho M. Implemente um menu com as seguintes opções para:

- exibir os valores do vetor;
- encontrar o maior valor deste vetor;
- encontrar os dois maiores valores do vetor, com apenas uma passagem;
- calcular a média dos valores do vetor;
- verificar se existem dois valores iguais no vetor retornando verdadeiro ou falso.

2) Faça um programa para trocar de posição dois elementos de um vetor usando uma variável auxiliar.

3) Para um vetor de n números, imprimir seus elementos na ordem inversa.

4) Faça um programa em C que gere aleatoriamente através de um laço *for()* um vetor com 10 elementos (números inteiros entre 0 e 9) sem que haja repetição de valores. Em seguida, imprima o vetor e informe a quantidade de tentativas que falharam (geraram números que já estavam no vetor).

5) Dado um vetor x_1, x_2, \dots, x_k de números inteiros, verifique se existem dois segmentos consecutivos iguais, isto é, se existem i e m tais que:

$$x_i, x_{i+1}, \dots, x_{i+m-1} = x_{i+m}, x_{i+m+1}, \dots, x_{i+2m-1}$$

Caso existam, imprima as posições i e m .

Ex:

Na seqüência 7, 9, 5, 4, 5, 4, 8, 6 temos dois segmentos consecutivos iguais com $i=3$ e $m=2$.

6) Escreva uma programa em C que aceite um vetor de números e retorne a mediana dos números no vetor:

- A mediana de um vetor de números é o elemento m do vetor, tal que metade dos números restantes no vetor é maior ou igual a m e metade é menor ou igual a m , se o número de elementos no vetor for **ímpar**;
- Se o número de elementos for **par** a mediana será a média dos dois elementos $m1$ e $m2$ tal que metade dos elementos restantes é maior ou igual a $m1$ e $m2$, e metade dos elementos restantes é menor ou igual a $m1$ e $m2$.

7) Escreva um programa em C que aceite um vetor de números e retorne a moda ou uma indicação de que a moda não existe. A moda de um vetor de números é o número m no vetor que é repetido com maior frequência. Se mais de um número for repetido com frequência máxima igual, não existirá uma moda.

Notas:

(i) Para a geração de números aleatórios utilize as funções *rand()* e *srand()* da biblioteca *stdio.h*

(ii) Para resolver o exercício 6 pode-se criar uma matriz $N \times 2$ para armazenar os valores que se repetem e a frequência.