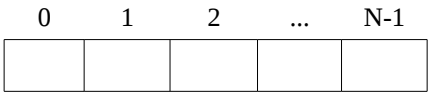
Disciplina: Linguagem de Programação II
Professora: Caroline Mazetto Mendes

Aula 8 - 28/03/17

1) Vetores (Deitel 2011, cap 6)

Um *array* consiste em uma coleção de dados de mesmo tipo, armazenados em endereços adjacentes de memória. *Arrays* unidimensionais são denominados vetores. Um vetor pode conter várias posições. Cada posição possui um índice associado, iniciando a partir do índice 0 até N-1, onde N é o tamanho do vetor.



Para declarar um vetor deve-se informar o tipo de dados, o identificador (nome) e o tamanho. Por exemplo, pode-se declarar um vetor do tipo inteiro chamado vetor com tamanho 50:

```
int vetor[50];          /* Declarando um vetor de int */
```

Um int ocupa 4 bytes de memória. Assim, ao encontrar esta declaração, o compilador aloca 4 * 50 bytes consecutivos de memória. Cada posição do vetor pode ser acessada informando o nome do vetor seguido pelo índice entre colchetes:

```
vetor[0] = 10;          /* Primeira posição do vetor */
vetor[49] = 30;         /* Última posição do vetor */
```

Podem ser declarados também vetores de outros tipos, como float, double e char.

Inicialização de vetores

Para inicializar um vetor durante a sua declaração, um valor deve ser especificado para cada cada posição. O valores devem ser informados entre chaves e são separados por virgula:

```
int idades[4] = {17, 20, 21, 18};              /* Inicializando um vetor de int */
float salarios[3] = {105.3, 321.5, 224.0};     /* Inicializando um vetor de float */
```

```
int erro[5];
erro = {2, 4, 6, 8, 10} ;          /* Isso está errado! */
```

Para inicializar ou preencher um vetor com valores após a sua declaração, é necessário atribuir o valor a cada posição do vetor. Para isso, pode-se utilizar uma estrutura de repetição para percorrer todas as posições do vetor. Por exemplo:

```
for (i=0; i<N; i++) {
        vetor[i] = 0;
}
```

Nesse exemplo, percorre-se o vetor da primeira até a última posição atribuindo o valor 0 a cada posição.

Limites do vetor

Quando um vetor é declarado com tamanho N, é alocado espaço em memória para N posições. Apesar de um vetor possuir um tamanho específico, é possível que um programa tente acessar endereços de memória que não pertencem ao vetor. Isso ocorre quando usa-se um índices que não estejam entre 0 e N-1. Nesse caso, o compilador não vai gerar nenhum aviso, e o programa pode não funcionar conforme o esperado.

Vetor como parâmetro de funções

Para passar um vetor como parâmetro (com todos os seus elementos) para uma função, basta realizar a chamada da função passando o nome do vetor. No protótipo da função, deve-se declarar o parâmetro correspondente como um sendo um vetor com o mesmo tamanho.

Exemplo 1. Preencher e imprimir um vetor com valores informados pelo usuário.

```
#include <stdio.h>

#define TAM 50

void preenche_vetor (int vetor[TAM], int num_elementos) {
        int i;

        printf("Entre com os elementos do vetor: \n");
        for (i=0; i<num_elementos; i++) {
                scanf("%d", &vetor[i]);
        }
}

void imprime_vetor (int vetor[TAM], int num_elementos) {
        int i;

        printf("Elementos do vetor: \n");
        for (i=0; i<num_elementos; i++) {
                printf("%d ",  vetor[i]);
        }
}

int main(void) {
        int vetor[TAM];
        int num_elementos;

        printf("Entre com o número de elementos do vetor:\n");
        scanf("%d", &num_elementos);

        preenche_vetor(vetor, num_elementos);
        imprime_vetor(vetor, num_elementos);

        return 0;
}
```

Código 1: Programa preenche um vetor com valores informados pelo usuário e imprime o vetor.

Informação importante: Uma variável vetor (informando apenas o nome) corresponde a um ponteiro que armazena o endereço de memória onde começa esse vetor. Portanto, a passagem de vetores como parâmetro para funções ocorre automaticamente por referência.

Considerações finais

Considere o seguinte trecho de código:

```
int vetorA[4] = {10, 20, 30, 40};
int vetorB[4];

vetorB = vetorA;                /* Isso está errado! */

if (vetorA == vetorB) {         /* Isso está errado! */
        printf("O vetores sao iguais.\n");
}
```

Para copiar vetores, deve-se percorrer todas as posições do vetor usando uma estrutura de repetição, copiando elemento por elemento. Da mesma forma, para comparar se dois vetores são iguais, deve-se comparar elemento por elemento.

2) Atividade Avaliativa 7

1. Faça um programa em C para realizar as seguintes operações:

a) Pedir para o usuário informar o número de elementos do vetor.
b) Preencher um vetor de números inteiros com valores informados pelo usuário.
c) Imprimir todos os elementos do vetor.
d) Calcular número de elementos ímpares do vetor.
e) Calcular a soma de todos os elementos do vetor.
f) Calcular o menor elemento do vetor.
g) Imprimir os resultados das operações dos itens (d), (e) e (f).

2. Faça um programa em C para gerar um relatório do estoque de uma loja. Considere um vetor em que as posições representam os produtos vendidos pela loja. Cada posição armazena a quantidade em estoque do *i-ésimo* produto. O programa deve realizar as seguintes operações:

a) Pedir para o usuário informar o número de produtos da loja.
b) Preencher um vetor com a quantidade em estoque de cada produto.
c) Calcular a quantidade total de produtos em estoque.
d) Calcular qual produto possui menor estoque.
e) Calcular qual produto possui maior estoque.
f) Calcular quais produtos estão com estoque vazio.
g) Imprimir o resultado das operações.

Obs.: A nota da atividade está condicionada à apresentação dos programas na aula em que foi proposta.