

Estruturas de dados homogêneas

Vetores

- Um vetor é uma sequência de vários valores do mesmo tipo, armazenados sequencialmente na memória, e fazendo uso de um mesmo nome de variável para acessar esses valores.
- Um vetor também pode ser entendido logicamente como uma lista de elementos de um mesmo tipo.
- Cada elemento dessa sequência pode ser acessado individualmente através de um índice dado por um nº inteiro.
- Os elementos são **indexados** de 0 até N-1, onde N é a quantidade de elementos no vetor.
- O valor de N também é chamado de dimensão ou tamanho do vetor.
- O vetor tem tamanho fixo durante a execução do programa, definido na declaração.
- Durante a execução não é possível aumentar ou diminuir o tamanho do vetor.
- A numeração dos índices começa em zero.

Declaração de vetores

- Obedece à mesma sintaxe da declaração de variáveis.
- A diferença está no uso de colchetes, que determina quantos elementos ele armazenará, ou seja, determina sua dimensão ou tamanho.

Sintaxe:

tipo variável[tamanho];

Exemplo:

int vetor[10];

- O tamanho precisa ser necessariamente um número inteiro e constante. Ele não pode ser resultado de uma expressão.

Acesso ao conteúdo de vetores

- Já vimos que a invocação é capaz de acessar um dado na memória.
- O tipo de valor retornado é igual ao tipo utilizado na declaração da variável.
- O nome da variável também é chamado de uma expressão de **referência de memória**.
- Com os vetores, temos uma nova expressão de referência de memória: o operador de índice “[]”.
- Ele utiliza uma referência de memória e um número inteiro.

- Ele retorna uma referência para o elemento correspondente ao índice.
- O tipo do valor retornado é do mesmo tipo da declaração do vetor.

Exemplo: vetor[0] = 3;

O programador deve decidir o tamanho do vetor na hora de implementação.

Int n; (tá errado)
scanf("%d", &n);
int vetor[n];

int vetor[1000]; (modo certo)

Conteúdo inicial de vetores

- Na declaração pode-se definir o valor inicial de cada elemento do vetor.
- A sintaxe é semelhante à declaração de uma variável comum com valor inicial, mas os elementos são listados entre as chaves { e }, e separado por vírgulas.

Sintaxe:

tipo variável[n] = {e₁, e₂, ..., e_{n-1}};

- como o tamanho do vetor pode ser inferido a partir da lista entre chaves, podemos omitir o tamanho do vetor.

Tipo variável[] = {e₁, e₂, ..., e_{n-1}};

Regras para acesso ao vetor

- Índices inválidos
- Copiar vetores
- Atribuir o valor de todos os elementos de uma só vez.

Exemplos:

1. Escreva um programa que leia 20 números inteiros, calcule a média aritmética dos valores e imprima na tela os elementos informados que sejam maiores que a média.

2. Escreva um programa que leia 10 números inteiros, em seguida, leia mais um número e verifique se este está no vetor. Caso esteja, imprima a posição onde o elemento se encontra no vetor.

```
Int main(){ //forma 1 de resolver
    char nome[100];
    int tam, i;
    printf("Digite um nome: ");
    fgets(nome, 100, stdin);
    for(i = 0; i<100 && nome[i] != \0, i++){
        tam++
    }
    return 0;
}
```

```
Int main(){ //forma 2 de resolver
    char nome[100];
    int tam, i;
    printf("Digite um nome: ");
    fgets(nome, 100, stdin);
    for(i = 0; i<100; i++){
        if(nome[i]=='\0')
            break
    }
    printf("%d\n", i);
    return 0;
}
```