



# MC-102 ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES IC-UNICAMP

## Aula 11 - Vetores

POR: EVANDRO CESAR BRACHT  
(SALA IC-86) 1S2005

### 1 Objetivos

Aprender o uso da estrutura de dados conhecida como vetores, declaração, inicialização, atribuição de valores e acesso ao conteúdo.

### 2 Motivação

Utilizar um grande número de variáveis relacionadas sem a necessidade de criar nomes diferentes.

### 3 Aula e Exemplos

Peça para os alunos fazer um programa que leia as notas de uma prova para uma classe de 5 alunos e então calcular a média da classe.

Exemplo de resposta:

```
#include <stdio.h>
int main(){
    int nota0,nota1,nota2,nota3,nota4,media;
    printf("Digite a nota do aluno 0: ");
    scanf("%d",&nota0);
    printf("Digite a nota do aluno 1: ");
    scanf("%d",&nota1);
    printf("Digite a nota do aluno 2: ");
    scanf("%d",&nota2);
    printf("Digite a nota do aluno 3: ");
    scanf("%d",&nota3);
    printf("Digite a nota do aluno 4: ");
    scanf("%d",&nota4);
    media = (nota0+nota1+nota2+nota3+nota4)/5;
```

```
    printf(‘‘média: %d\\n’’,media);  
}
```

Agora resolva o mesmo problema para uma turma de 50 alunos e para uma escola com 2000 alunos.

Um **vetor** é um tipo de dado usado para representar uma certa quantidade de variáveis de valores homogêneos.

### 3.1 A declaração

A sintaxe em C para a criação de um vetor é:

*tipo identificador[número de variáveis];*

Onde

- *tipo* é o tipo das variáveis que devem ser criados. Exemplo: *int*, *char*, *float*, entre outros;
- *identificador* é o nome que será utilizado para referenciar o conjunto de variáveis;
- *número de variáveis* é o número de variáveis que será criado.

Um exemplo de declaração de vetor em C:

```
int notas[5];
```

### 3.2 Referenciando um elemento do vetor

Após declarar um vetor precisamos de um modo de referenciar seus elementos individualmente. Isto é feito através do numero entre colchetes seguindo o nome do vetor.

Note a diferença nas linhas que se segue.

```
int notas[5];  
notas[2] = 85;
```

Na primeira linha temos a declaração do vetor *notas* do tipo inteiro contendo 5 posições. Já na segunda linha temos a atribuição do valor 85 a terceira posição do vetor *notas*.

Isto mesmo, não esta errado não, é a terceira posição do vetor *notas*. Os elementos de um vetor são sempre numerados por índices iniciados por 0(zero). Ou seja, as 5 posições do vetor *notas* são acessadas através dos índices 0, 1, 2, 3, e 4. Além disso, podemos referenciar posições do vetor utilizando variáveis inteiras. O exemplo que segue tem o mesmo resultado que o exemplo anterior.

```
int notas[5];  
int i = 2;  
notas[i] = 85;
```

### 3.3 Armazenando dados em um vetor

Em seguida temos um exemplo de como fazer a leitura dos dados de um vetor.

```
for(i=0; i<5; i++){
    printf("Digite a nota do aluno %d: ",i);
    scanf("%d",&notas[i]);
}
```

### 3.4 Lendo dados de um vetor

Em seguida temos um exemplo de como é feita a leitura dos dados já armazenados em um vetor.

```
soma = 0;
for(i=0; i<5; i++)
    soma = soma + notas[i];
printf("Media das notas: %d.",soma/5);
```

### 3.5 Lendo um número desconhecido de elementos

Supomos agora que o número de notas que deve ser lido não é conhecido, a única informação que temos é que nenhuma classe pode ter mais do que 40 alunos, ou seja, temos no máximo 40 notas.

O programa exemplo seguinte será executado até que uma nota menor ou igual a 0 (zero) seja digitada ou atinja o limite de 40 notas.

```
#include <stdio.h>
#define LIM 40
int main(){
    float notas[LIM],soma=0.0;
    int i=0;
    do {
        if(i>=LIM){
            printf("Buffer lotado.\n");
            i++;
            break; // sai do laço do-while
        }
        printf("Digite a nota do aluno %d: ",i);
        scanf("%f",&notas[i]);
        if(notas[i]>0)
            soma+=notas[i];
    } while (notas[i++]>0);
    printf("Media das notas: %.2f\n",soma/(i-1));
    return 0;
}
```

### 3.6 Inicializando vetores

Assim como em variáveis podemos atribuir um valor inicial a um vetor, esta inicialização deve ser feita no momento da criação do vetor. O trecho de código a seguir exemplifica a inicialização do vetor *moedas* com os valores 50 na posição 0, 25 na posição 1, 10 na posição 2, 5 na posição 3 e 1 na posição 4.

```
int moedas[5]={50,25,10,5,1};
```

### 3.7 Vetores e Strings

Em C uma **string** é um vetor do tipo **char** terminada pelo caracter null (`'\0'`). No exemplo que segue temos a criação de um vetor de caracteres que pode ser usado para armazenar caracteres independentes ou uma string.

```
char s[20];
```

Podemos armazenar informações neste vetor cada caracter individualmente , como mostra o segmento de código a seguir

```
for(i=0; i<20; i++){
    printf("Digite o caracter %d: ",i);
    scanf("%c",&s[i]);
}
```

ou podemos armazenar todos os caracteres como uma string.

```
printf("Digite a sequência de caracteres: ");
scanf("%s",s);
```

## 4 Exemplos e/ou Exercícios

1. A declaração do vetor esta correto? Qual o erro?

```
int vetor(25);
```

**R:** Está errada, o correto é `int vetor[25];`, usar `[ ]` no lugar de `( )`.

2. Qual o elemento da matriz referenciado por pela expressão?

```
vetor[4]
```

**R:** Quinto elemento.

3. Ler 30 notas, calcular a média e mostrar as notas que estão acima da média e quais são os alunos que tem notas acima da média. **R:**

```

#include <stdio.h>
#define LIM 30
int main(){
    float notas[LIM],soma=0.0, media;
    int i=0;
    for (i=1; i<LIM; i++) {
        printf("Digite a nota do aluno %d: ",i);
        scanf("%f",&notas[i]);
        soma+=notas[i];
    }
    media = soma/LIM;
    printf("Media das notas: %.2f e \n",media);
    for (i=1; i<LIM; i++) {
        if (notas[i]>= media)
            printf("o aluno %d tem nota %.2f maior que a média.",i,notas[i]);
    }
    return 0;
}

```

4. Ler as notas de 20 alunos e armazena-las em um vetor, então diga se existe algum aluno que tenha uma nota  $x$  escolhida pelo usuário. **R:**

```

#include <stdio.h>

int main()
{
    float nota[21],x; /* vetor criado com uma posição a mais */
    int i;

    printf("Entre com 10 notas\n");
    for (i=0; i < 10; i++) {
        scanf("%f",&nota[i]);
    }

    while(1) {
        printf("Digite a nota procurada ou -1 para sair do programa\n");
        scanf("%f",&x);

        if (x== -1.0)
            break;

        /* busca linear */

        nota[10] = x; /* elemento sentinela */
        i = 0;
        while (nota[i] != x) /* busca com sentinela */
            i++;
    }
}

```

```
    if (i < 10)
        printf("nota %5.2f encontrada na posição %d\n",nota[i],i);
    else
        printf("nota %5.2f não encontrada\n",x);
}

return 0;
}
```

## 5 Referências

Estas aulas foram baseadas em:

- Notas de aula do **prof. Alexandre Falcão**  
(<http://www.dcc.unicamp.br/~afalcao/mc102/notas-aula.pdf>)
- Apostila do **prof. Flávio Keidi Miyazawa**
- Livro *Treinamento em Linguagem C, Curso Completo, Módulo 1.* de Victorine, Viviane e Mizrahi, Makron books.