



# VETORES

PROGRAMAÇÃO APLICADA

Isadora Maria Gonçalves Ramos

Monte Carmelo/ 2018

# IMAGINE....

- Imagine que você foi contratado para criar um programa em C para uma escola. Nesse programa você tem que armazenar as notas dos alunos, nomes, médias, nome dos pais, faltas e tudo mais.
- E aí? Vai declarar quantos inteiros pra armazenar as notas? Centenas? Milhares?  
E quantos caracteres para armazenar esses nomes?  
E quantos floats para armazenar as notas e médias, de cada matéria, para cada aluno?



# COMPLICADO NÉ!?

- É quase impossível realizar isso...

... quase.



- Por isso, para facilitar a vida das pessoas, existe a programação.
- Você pode fazer isso tudo em poucos minutos, pois programa em C e tem total domínio de sua máquina, você não perderá tempo declarando inúmeras variáveis, fará com que o computador faça isso pra você.



A SOLUÇÃO SÃO...

## ■ Vetores



# O QUE SÃO VETORES?

- O vetor é uma estrutura de dados indexada, que pode armazenar uma determinada quantidade de valores do mesmo tipo.
- Os dados armazenados em um vetor são chamados de itens do vetor.
- Para localizar a posição de um item em um vetor usamos um número inteiro denominado índice do vetor.
- Os **Vetores** são matrizes de uma só dimensão.
- Também chamados ***arrays***

# RESUMINDO

- 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 desta sequência pode ser acessado individualmente através de um índice dado por um número inteiro. Os elementos são indexados de 0 até  $n-1$ , onde  $n$  é a quantidade de elementos do 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. Note que a numeração começa em zero, e não em um. Essa é uma fonte comum de erros. A Figura 1 ilustra um vetor com 10 elementos, denominados  $v_0, v_1, \dots, v_9$ , todos eles de tipo `int`.

`int[10]`

$v_0$	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$	$v_6$	$v_7$	$v_8$	$v_9$
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

**Figura 1 – Exemplo de vetor com 10 elementos**



# VANTAGEM DE UTILIZAÇÃO DO VETOR

- Facilidade de manipular um grande conjunto de dados do mesmo tipo declarando-se apenas uma variável.

# DECLARAÇÃO DE VETORES

- `tipo nome_var[tamanho];`

# ACESSANDO UM VETOR

- Os vetores são acessados através de índices colocados entre colchetes.
- O índice do primeiro elemento do vetor é 0 (ZERO).
- EXEMPLOS:

```
int amostra[10]; /* vetor de 10 inteiros */
```

```
amostra[0] = 2; /* primeiro elemento */
```

```
amostra[9] = 7; /* último elemento
```

# IMAGINE O SEGUINTE PROBLEMA:

- Faça um programa que leia as notas de uma turma de cinco estudantes e depois imprima as notas que são maiores do que a média da turma.
- Um algoritmo para esse problema poderia ser o mostrado a seguir.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    float n1,n2,n3,n4,n5;
    printf("Digite a nota de 5 estudantes: ");
    scanf("%f",&n1);
    scanf("%f",&n2);
    scanf("%f",&n3);
    scanf("%f",&n4);
    scanf("%f",&n5);
    float media = (n1+n2+n3+n4+n5)/5.0;
    if(n1 > media) printf("nota: %f\n",n1);
    if(n2 > media) printf("nota: %f\n",n2);
    if(n3 > media) printf("nota: %f\n",n3);
    if(n4 > media) printf("nota: %f\n",n4);
    if(n5 > media) printf("nota: %f\n",n5);

    return 0;
}
```

Como foi visto em sala.

- O algoritmo anterior apresenta uma solução possível para o problema apresentado. Porém, essa solução é inviável para grandes quantidades de alunos.
- Imagine se tivéssemos de processar as notas de 100 alunos.
- Para 100 alunos, precisamos de:
- **Uma variável para armazenar a nota de cada aluno.**  
**100 variáveis;**
- **Um comando de leitura para cada nota.**  
**100 scanf();**
- **Um somatório de 100 notas.**
- **Um comando de teste para cada aluno.**
- **100 comandos if;**
- **Um comando de impressão na tela para cada aluno.**
- **100 printf() ;**

# RESOLVENDO...

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    float notas[100];
    int i;
    printf("Digite as notas dos estudantes\n");
    for(i = 0; i < 100; i++) {
        printf("Nota do estudante %d:", i);
        scanf("%f", &notas[i]);
    }
    float media = 0;
    for(i = 0; i < 100; i++)
        media = media + notas[i];
    media = media / 100;

    for(i = 0; i < 100; i++)
        if(notas[i] > media)
            printf("Notas: %f\n", notas[i]);

    return 0;
}
```

## EXEMPLO:

- Leia 10 números inteiros positivos e armazene em um vetor. Em seguida determine e imprima o maior elemento no vetor.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  main()
4  {
5
6      int v[10], i, maior = 0;
7      for (i=0; i<10; i++)
8          scanf("%d", &v[i]);
9      for (i=0; i<10; i++)
10         if (v[i] > maior)
11             maior = v[i];
12     printf("maior = %d\n", maior);
13
14
15     return 0;
16 }
17
```

- Agora que você aprendeu o que é, e para que serve um vetor, é só colocar em prática e aplicar!





# REFERÊNCIAS

- Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados São Paulo: Forbellone, André Luiz Villar - MAKRON, 1993.
- MAITELLI, André, Apostila do Curso de CLP - Engenharia Elétrica, UFRN, 2002.
- STROUSTRUP, Bjarne. The C++ Programming Language - Third Edition. Addison-Wesley, 1997.