

Vetores

Algoritmos e Programação de Computadores – ABI/LFI/TAI



**INSTITUTO
FEDERAL**
Brasília

Prof. Daniel Saad Nogueira
Nunes

IFB – Instituto Federal de Brasília,
Campus Taguatinga



Sumário

- 1 Introdução
- 2 Vetores
- 3 Exemplos



Sumário

1 Introdução



Introdução

- Imagine o seguinte problema: dados 100 valores inteiros lidos do teclado, verificar se todos são distintos.
- Como resolvê-lo?
- Poderíamos declarar **cem** variáveis inteiras e realizar uma comparação par a par.



Introdução

```
1  #include <stdio.h>
2
3  int main(void){
4      int numero_1, numero_2, ..., numero_100;
5      printf("Digite o primeiro valor: ");
6      scanf("%d",&numero_1);
7      ...
8      printf("Digite o centésimo valor: ");
9      scanf("%d",&numero_100);
10     // realiza comparações
11     ...
12     return 0;
13 }
```



Introdução

- Esta solução, apesar de funcionar, é extremamente massante e nada elegante.
- Felizmente, na linguagem C podemos declarar uma coleção de variáveis de um mesmo tipo.
- Mecanismo: **vetores!**



Sumário

2 Vetores



Vetores

- Os vetores fornecem uma maneira de declarar uma coleção de variáveis do mesmo tipo sob o mesmo identificador.
- Através do identificador, podemos acessar qualquer variável de uma maneira bem simples.



Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- Cuidados
- Inicialização



Vetores: Sintaxe

- Para declarar um vetor de tamanho `tam` de um determinado `tipo`, utilizamos a seguinte sintaxe: `tipo nome_vetor[tam];`
- Por exemplo, declaração de um vetor de inteiros, chamado `idades`, de tamanho 100: `int idades[100];`
- Declaração de um vetor de números ponto-flutuante, de precisão dupla, chamado `salarios`, de tamanho 50: `double salarios[50];`



Vetores: Sintaxe

- Para acessar o i -ésimo valor do vetor, utilizamos o operador `[i]`.
- O primeiro valor do vetor está na posição 0.
- O último valor do vetor, está na posição $n - 1$, em que n é o tamanho do vetor.



Vetores: Sintaxe

- Para acessar o terceiro valor do vetor `salarios`: `salarios[2];`
- Para acessar o último valor do vetor `idades`: `idades[99];`
- Para acessar o primeiro valor do vetor `idades`: `idades[0];`



Sumário

2 Vetores

- Sintaxe
- **Organização**
- Utilização
- Cuidados
- Inicialização



Vetores: Organização

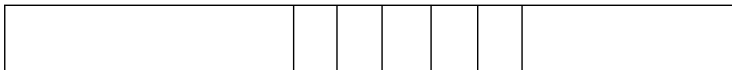
- Ao declarar um vetor de um determinado tamanho n , temos a presença de n espaços consecutivos de memória.
- Exemplo para um vetor de tamanho 5:

```
int vetor[5];
```

vetor

0 1 2 3 4

Memória





Vetores: Organização

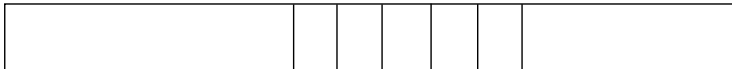
- Ao declarar um vetor de um determinado tipo T de tamanho n , temos a presença de n espaços consecutivos de memória, cada um com tamanho igual ao exigido pelo tipo T.
- Exemplo para um vetor de inteiros de tamanho 5:

```
int vetor[5];
```

```
vetor
```

1000 1004 1008 1012 1016

Memória





Sumário

2 Vetores

- Sintaxe
- Organização
- **Utilização**
- Cuidados
- Inicialização



Vetores: Utilização

- Prosseguindo com o nosso problema inicial, vamos construir um programa que leia 100 valores inteiros e verifique se todos os valores são distintos.



Vetores: Utilização

Estratégia

- Ler os cem valores em um vetor de inteiros.
- Realizar comparações par a par entre os elementos e através de uma variável indicadora, verificar:
 - ▶ Se existe algum par de elementos iguais, então a resposta é falsa.
- Após todas as comparações, se foi verificado que não há par de elementos iguais, então a resposta é verdadeira.



Vetores: Utilização

Declaração

```
4  int i, j;  
5  int numeros[100];  
6  int resposta = 1;
```



Vetores: Utilização

Leitura

```
7   for (i = 0; i < 100; i++) {  
8       printf("Digite o elemento %d: ", i);  
9       scanf("%d", &numeros[i]);  
10  }
```



Vetores: Utilização

Comparação

```
11     for (i = 0; i < 100 && resposta; i++) {  
12         for (j = i + 1; j < 100 && resposta; j++) {  
13             if (numeros[i] == numeros[j])  
14                 resposta = 0;  
15         }  
16     }
```



Vetores: Utilização

Checagem

```
if (resposta) {  
    printf("Todos os elementos são distintos.\n");  
}  
else {  
    printf("Existem elementos repetidos.\n");  
}
```



Vetores: Utilização

Programa Completo

```
1 #include <stdio.h>
2
3 int main(void) {
4     int i, j;
5     int numeros[100];
6     int resposta = 1;
7     for (i = 0; i < 100; i++) {
8         printf("Digite o elemento %d: ", i);
9         scanf("%d", &numeros[i]);
10    }
11    for (i = 0; i < 100 && resposta; i++) {
12        for (j = i + 1; j < 100 && resposta; j++) {
13            if (numeros[i] == numeros[j])
14                resposta = 0;
15        }
16    }
```



Vetores: Utilização

Programa Completo

```
17     if (resposta) {
18         printf("Todos os elementos são distintos.\n");
19     }
20     else {
21         printf("Existem elementos repetidos.\n");
22     }
23     return 0;
24 }
```




Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- **Cuidados**
- Inicialização



Vetores: Cuidados

Out of Bounds

- Um vetor na C tem um tamanho fixo.
- É um erro de lógica tentar acessar posições que não pertencem ao vetor.
- As posições válidas de um vetor de tamanho n estão no intervalo $[0, n - 1]$.
- O acesso a alguma posição inválida, como -1 ou n , por exemplo, pode ocasionar uma **falha de segmentação**.



Vetores: Cuidados

Out of Bounds

- O compilador não verifica isso para você.
- É total responsabilidade do programador controlar o acesso aos elementos de um vetor.



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=1;i<=5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=1;i<=5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=1;i<=5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=1;i<=5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Programa Corrigido

```
1  #include <stdio.h>
2
3  int main(void){
4      int vetor[5];
5      int i;
6      for(i=0;i<5;i++){
7          scanf("%d",&vetor[i]);
8      }
9      printf("Os elementos digitados foram: ");
10     for(i=0;i<5;i++){
11         printf("%d ",vetor[i]);
12     }
13     printf("\n");
14     return 0;
15 }
```



Vetores: Cuidados

Declaração

- Como os vetores tem tamanho fixo, é importante escolher o seu tamanho adequadamente.
- Não se pode declarar um tamanho menor do que o necessário.



Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void) {
4      int vetor[5];
5      int n;
6      int i;
7      printf("Digite o número de valores que deseja ler: ");
8      scanf("%d", &n);
9      for (i = 0; i < n; i++) {
10         scanf("%d", &vetor[i]);
11     }
12     printf("Os elementos digitados foram: ");
13     for (i = 0; i < n; i++) {
14         printf("%d ", vetor[i]);
15     }
16     printf("\n");
17     return 0;
18 }
```




Vetores: Cuidados

Encontre o Erro

```
1  #include <stdio.h>
2
3  int main(void) {
4      int vetor[5];
5      int n;
6      int i;
7      printf("Digite o número de valores que deseja ler: ");
8      scanf("%d", &n);
9      for (i = 0; i < n; i++) {
10         scanf("%d", &vetor[i]);
11     }
12     printf("Os elementos digitados foram: ");
13     for (i = 0; i < n; i++) {
14         printf("%d ", vetor[i]);
15     }
16     printf("\n");
17     return 0;
18 }
```



Vetores: Cuidados

Programa Corrigido

```
1  #include <stdio.h>
2  int main(void) {
3      int vetor[5];
4      int n;
5      int i;
6      printf("Digite o número de valores que deseja ler: ");
7      scanf("%d", &n);
8      if (n > 5) {
9          printf("Número de valores alterado para 5.\n");
10         n = 5;
11     }
12     for (i = 0; i < n; i++) {
13         scanf("%d", &vetor[i]);
14     }
15     printf("Os elementos digitados foram: ");
16     for (i = 0; i < n; i++) {
17         printf("%d ", vetor[i]);
18     }
19     printf("\n");
20     return 0;
21 }
```



Vetores: Cuidados

Declaração

- Como os vetores tem tamanho fixo, é importante escolher o seu tamanho adequadamente.
- Não se pode declarar um tamanho menor do que o necessário.



Vetores: Cuidados

- Suponha as seguintes declarações:

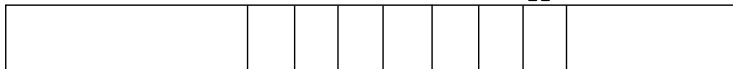
```
1  int n;  
2  int vetor[5];  
3  int x;
```

- Imagine que as variáveis estejam dispostas da seguinte forma na memória:

```
int vetor[5];
```

```
      vetor  
      n 0 1 2 3 4 x
```

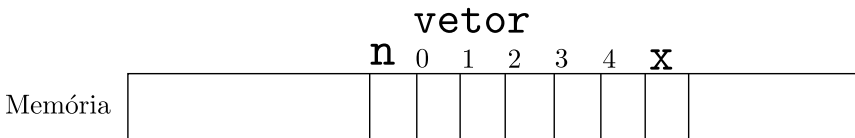
Memória





Vetores: Cuidados

```
int vetor[5];
```



- O que acontece se fizermos `vetor[5] = -2;`
- O que acontece se fizermos `vetor[-1] = 0;`
- Estaremos manipulando, indevidamente, as variáveis `n` e `x`.
- Mesmo que o seu programa não aborte, ele terá um *bug* difícil de detectar por conta do mal uso do vetor.



Sumário

2 Vetores

- Sintaxe
- Organização
- Utilização
- Cuidados
- Inicialização



Inicialização

- Em alguns casos pode-se desejar criar um vetor já inicializado com alguns elementos.
- Isso pode ser feito de maneira simples em C:

```
int vetor[] = {1,2,3,4,5};
```
- O tamanho do vetor é inferido automaticamente e é opcional colocá-lo neste tipo de declaração.



Inicialização

```
1  #include <stdio.h>
2
3  int main(void){
4      int i;
5      int vetor_int[] = {1,2,3,4,5};
6      float vetor_float[] = {0.5,2.3,1.1,2.4,3.9,-1.4};
7      for(i=0;i<5;i++){
8          printf("%d ",vetor_int[i]);
9      }
10     printf("\n");
11     for(i=0;i<6;i++){
12         printf("%.1f ",vetor_float[i]);
13     }
14     printf("\n");
15     return 0;
16 }
```




Sumário

3 Exemplos



Exemplos

Produto Escalar

- Dado dois vetores de números ponto-flutuante, precisão dupla, computar o seu produto escalar.
- Sejam V_1 e V_2 , dois vetores de tamanho $n \leq 10$, o produto escalar é definido como:

$$\sum_{i=0}^{n-1} V_1[i] \cdot V_2[i]$$

- A soma dos produtos de cada elemento do primeiro vetor com o elemento na posição equivalente do segundo vetor.



Exemplos

Estratégia

- A estratégia aqui é utilizar uma variável acumuladora para registrar o produto escalar a cada iteração do laço.



Exemplos

Leitura

```
8 scanf("%d", &n);
9 printf("Leitura do primeiro vetor.\n");
10 for (i = 0; i < n; i++) {
11     printf("v1[%d] = ", i);
12     scanf("%lf", &v1[i]);
13 }
14 printf("Leitura do segundo vetor.\n");
15 for (i = 0; i < n; i++) {
16     printf("v2[%d] = ", i);
17     scanf("%lf", &v2[i]);
18 }
```



Exemplos

Cálculo

```
19     for (i = 0, soma = 0.0; i < n; i++) {  
20         soma += v1[i] * v2[i];  
21     }  
22     printf("O produto escalar é %.2f.\n",soma);
```



Exemplos

Programa Completo

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i, n;
5      double soma;
6      double v1[10], v2[10];
7      printf("Digite o tamanho dos vetores: ");
8      scanf("%d", &n);
9      printf("Leitura do primeiro vetor.\n");
10     for (i = 0; i < n; i++) {
11         printf("v1[%d] = ", i);
12         scanf("%lf", &v1[i]);
13     }
```



Exemplos

Programa Completo

```
14  printf("Leitura do segundo vetor.\n");
15  for (i = 0; i < n; i++) {
16      printf("v2[%d] = ", i);
17      scanf("%lf", &v2[i]);
18  }
19  for (i = 0, soma = 0.0; i < n; i++) {
20      soma += v1[i] * v2[i];
21  }
22  printf("O produto escalar é %.2f.\n",soma);
23  return 0;
24 }
```



Exemplos

Exemplo

- Dado um vetor de inteiros de tamanho 10 e um inteiro C , encontrar dois elementos distintos do vetor cuja multiplicação é C e imprimi-los.
- Caso não haja esse par de elementos, uma mensagem deverá ser informada ao usuário.



Exemplos

Estratégia

- A estratégia aqui é checar todos os pares $(i, j), i \neq j$, de elementos do vetor e verificar se existe algum cuja multiplicação é C .
- Usamos dois laços aninhados e uma variável indicadora para verificar se o par já foi encontrado.
- Caso não haja tal par, uma mensagem é informada.



Exemplos

Leitura

```
7   for (i = 0; i < 10; i++) {
8       printf("v[%d] = ", i);
9       scanf("%d", &v[i]);
10      }
11      printf("Digite o valor de c: ");
12      scanf("%d", &c);
```



Exemplos

Cálculo

```
13     int encontrado = 0;
14     for (i = 0; i < 10 && !encontrado; i++) {
15         for (j = i + 1; j < 10 && !encontrado; j++) {
16             if (v[i] * v[j] == c) {
17                 encontrado = 1;
18                 printf("%d * %d = %d\n", v[i], v[j], c);
19             }
20         }
21     }
22     if (!encontrado) {
23         printf("Não existe um par de elementos que multiplicam %d", c);
24     }
```



Exemplos

Programa Completo

```
1  #include <stdio.h>
2
3  int main(void) {
4      int i, j;
5      int v[10];
6      int c;
7      for (i = 0; i < 10; i++) {
8          printf("v[%d] = ", i);
9          scanf("%d", &v[i]);
10     }
11     printf("Digite o valor de c: ");
12     scanf("%d", &c);
```



Exemplos

Programa Completo

```
13  int encontrado = 0;
14  for (i = 0; i < 10 && !encontrado; i++) {
15      for (j = i + 1; j < 10 && !encontrado; j++) {
16          if (v[i] * v[j] == c) {
17              encontrado = 1;
18              printf("%d * %d = %d\n", v[i], v[j], c);
19          }
20      }
21  }
22  if (!encontrado) {
23      printf("Não existe um par de elementos que multiplicam %d", c);
24  }
25  return 0;
26  }
```