

BCC 201 - Introdução à Programação I

Estruturas Homogêneas (Vetores)

Guillermo Cámara-Chávez
UFOP

Introdução I

Como armazenar 3 notas?

```
using namespace std;
int main()
{
    float nota1, nota2, nota3;
    cout << "Nota do aluno 1: \n";
    cin >> nota1;
    cout << "Nota do aluno 2: \n";
    cin >> nota2;
    cout << "Nota do aluno 3: \n";
    cin >> nota3;
    return 0;
}
```

Introdução II

Como armazenar 100 notas?

```
using namespace std;
int main()
{
    float nota1, nota2, nota3, /*...*/ nota100;
    cout << "Nota do aluno 1: \n";
    cin >> nota1;
    cout << "Nota do aluno 2: \n";
    cin >> nota2;
    /*...*/
    cout << "Nota do aluno 100: \n";
    cin >> nota100;
    return 0;
}
```

...declaramos 100 variáveis?

Não, torna-se inviável

Introdução III

Deseja-se calcular a média de 10 notas.

Introdução IV

```
using namespace std;
int main(){
    double nota , media , acum = 0;
    int i;
    for (i = 0; i < 10; i++){
        cout << "\n Inserir nota: ";
        cin >> nota;
        acum += nota;
    }
    media = acum / 10;
    cout << "\n A media é " << media;
    return 0;
}
```

Introdução V

Deseja-se calcular quantas notas, de um total de 10, são maiores que a média.

Introdução VI

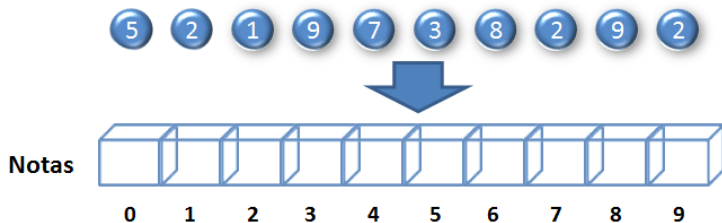
Sem salvar cada uma das notas, não é possível determinar quantas delas são maiores que a média

Introdução VII

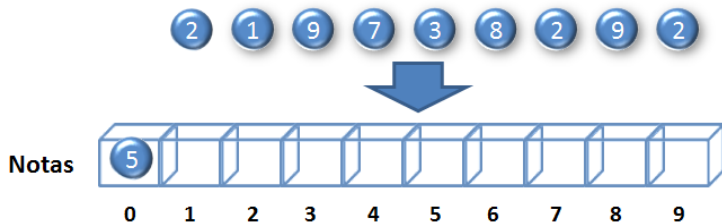
Em qual variável está a oitava nota?



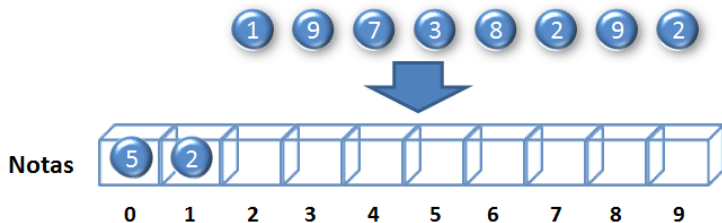
Introdução VIII



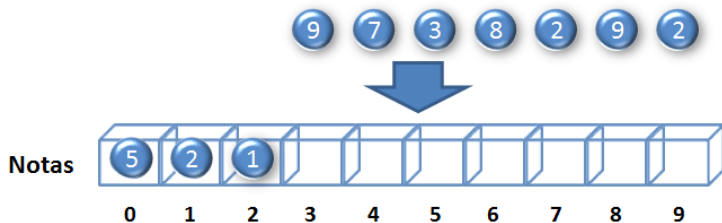
Introdução IX



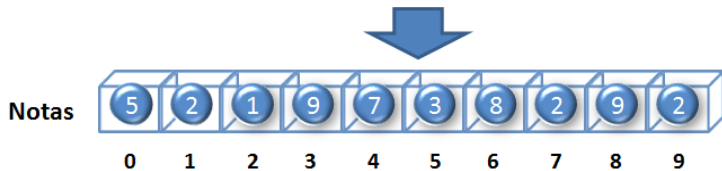
Introdução X



Introdução XI



Introdução XII



Introdução XIII

Em qual variável está a oitava nota?

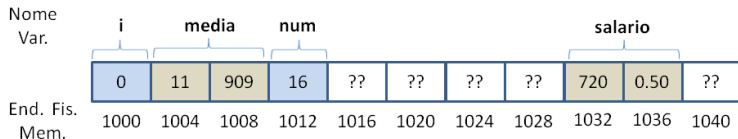


Introdução XIV

- ▶ Um vetor é
 - ▶ uma variável composta homogênea **unidimensional**,
 - ▶ formada por uma sequências de variáveis, todas do mesmo **tipo de dados**
 - ▶ com o mesmo identificador (mesmo nome) e,
 - ▶ alocadas sequencialmente na memória.

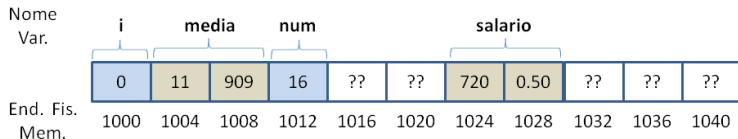
Introdução XV

```
float salario;  
int i = 0;  
float media = 11909;  
int num;  
...  
num = 16;  
salario = 7200.50;  
int vet[4]; // ok, existe espaço
```



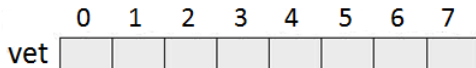
Introdução XVI

```
float salario;  
int i = 0;  
float media = 11909;  
int num;  
...  
num = 16;  
salario = 7200.50;  
int vet[4]; // não existe espaço
```



Introdução XVII

- ▶ Uma vez que as variáveis que compõem o vetor têm o mesmo nome, o que distingue cada um delas é um **índice**, que referencia sua localização dentro da estrutura
- ▶ Exemplo



Introdução XVIII



Declaração de um vetor I

```
<tipo> identificador [<número de posições>;
```

- ▶ Primeiro o tipo de dado: int, float, double, ...
- ▶ Segundo o nome da variável: usando as mesmas convenções de uma variável comum. (array, vetor, variavelDeNumeros, ...)
- ▶ E por fim, o tamanho necessário do vetor escrito entre colchetes: [5], [10], [3] ...

Declaração de um vetor II

- ▶ A primeira posição de um vetor tem índice 0.
- ▶ A última posição de um vetor tem índice $\langle \text{número de posições} \rangle - 1$.
- ▶ O sistema de execução não verifica se um dado acesso está dentro dos limites de um vetor.

Declaração de um vetor III

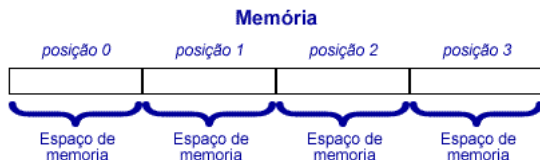
- ▶ Todo vetor é um espaço linear **dividido** de acordo com o **tamanho** que declaramos.
- ▶ Se declaramos `int array[4]`, na memória é representado da seguinte forma:



- ▶ Uma única variável com 4 espaços

Declaração de um vetor IV

- ▶ Para acessar um local específico basta indicar entre colchetes a posição desejada
- ▶ Aquela posição recebe o nome de índice.
- ▶ O índice sempre inicia em 0.
- ▶ Um vetor de tamanho 20 vai de 0 a 19.



Declaração de um vetor V

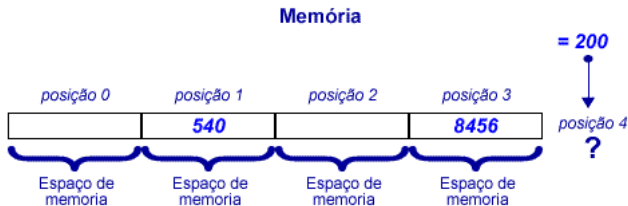
- ▶ Atribuir os valores 540 na posição 1 e 8456 na posição 3.

```
int array[4];  
array[1] = 540;  
array[3] = 8456;
```



Declaração de um vetor VI

- ▶ O limite do vetor é sempre seu tamanho menos 1.
- ▶ No exemplo anterior o vetor é de tamanho 4, a posição máximo é 3 ($4 - 1 = 3$).
- ▶ Se um valor for atribuído fora dos limites do vetor ocorrerá um erro.
- ▶ Não é possível fazer `array[4] = 200`.

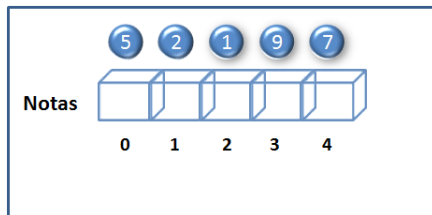


Exemplo I


Inserir 5 notas em um vetor, depois disso visualizar as notas inseridas.

Exemplo I

```
int main(){
    int j, notas[5];
    //Inserção de dados
    for (j = 0; j < 5; j++){
        cout<<"Inserir nota"
            <<j+1;
        cin >> notas[j];
    }
    //Visualização de dados
    for (j = 0; j < 5; j++)
        cout << notas[j];
    return 0;
}
```



Exemplo I


<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<< j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre>	<p><u>Simulação</u></p> <p>notas</p>  <p>0 1 2 3 4</p>

j=0


notas




Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<< j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 _</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j=0</p>	<p>notas</p>  <p>0 1 2 3 4</p>


Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<< j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j=0 notas[0] = 5</p>	<p>notas</p>  <p>0 1 2 3 4</p>


Exemplo I

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<< j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 1</p>	<p>notas</p>  <p>0 1 2 3 4</p>

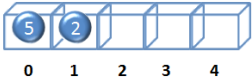
Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : _</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 1</p>	<p>notas</p>  <p>0 1 2 3 4</p>

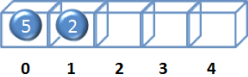
Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<< j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>j= 1 notas[1] = 2</p></div>	<p>notas</p>  <p>0 1 2 3 4</p>


Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 2</p>	<p>notas</p>  <p>0 1 2 3 4</p>


Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 :</p> <p><u>Estado das variáveis</u></p> <p>j= 2</p> <p><u>Simulação</u></p> <p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 2 notas[2] = 1</p>	<p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1</p> <p><u>Estado das variáveis</u></p> <div data-bbox="175 762 519 847" style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;">j= 3</div> <p><u>Simulação</u></p> <p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 1

Código

```
int main()
{
    int j, notas[5];
    //Inserção de dados
    for (j = 0; j < 5; j++){
        cout << "Inserir nota"<<j+1;
        cin >> notas[j];
    }
    //Visualização de dados
    for (j = 0; j < 5; j++)
        cout << notas[j] << endl;
    return 0;
}
```



Janela de execução

Inserir nota 1 : 5
Inserir nota 2 : 2
Inserir nota 3 : 1
Inserir nota 4 :

Estado das variáveis

j=3

Simulação



Exemplo 1

Código

```
int main()
{
    int j, notas[5];
    //Inserção de dados
    for (j = 0; j < 5; j++){
        cout << "Inserir nota"<<j+1;
        cin >> notas[j];
    }
    //Visualização de dados
    for (j = 0; j < 5; j++)
        cout << notas[j] << endl;
    return 0;
}
```



Janela de execução

Inserir nota 1 : 5
Inserir nota 2 : 2
Inserir nota 3 : 1
Inserir nota 4 : 9

Estado das variáveis

j= 3
notas[3] = 9

Simulação



Exemplo 1

Código

```
int main()
{
    int j, notas[5];
    //Inserção de dados
    for (j = 0; j < 5; j++){
        cout << "Inserir nota"<<j+1;
        cin >> notas[j];
    }
    //Visualização de dados
    for (j = 0; j < 5; j++)
        cout << notas[j] << endl;
    return 0;
}
```



Janela de execução

Inserir nota 1 : 5
Inserir nota 2 : 2
Inserir nota 3 : 1
Inserir nota 4 : 9


Estado das variáveis

j= 4


Simulação




Exemplo 1

<u>Código</u>	<u>Janela de execução</u>										
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1 Inserir nota 4 : 9 Inserir nota 5 : _</p>										
<u>Estado das variáveis</u>	<u>Simulação</u>										
<p>j= 4</p>	<p>notas</p>  <table border="1"><tr><td>5</td><td>2</td><td>1</td><td>9</td><td></td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table>	5	2	1	9		0	1	2	3	4
5	2	1	9								
0	1	2	3	4							


Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1 Inserir nota 4 : 9 Inserir nota 5 : 7</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<div style="border: 1px solid black; padding: 5px; width: fit-content;"><p>j= 4 notas[4] = 7</p></div>	<p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>→</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1 Inserir nota 4 : 9 Inserir nota 5 : 7</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 5</p>	<p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 1

<u>Código</u>	<u>Janela de execução</u>
<pre>int main() { int j, notas[5]; //Inserção de dados for (j = 0; j < 5; j++){ cout << "Inserir nota"<<j+1; cin >> notas[j]; } //Visualização de dados for (j = 0; j < 5; j++) cout << notas[j] << endl; return 0; }</pre> <p>➔</p>	<p>Inserir nota 1 : 5 Inserir nota 2 : 2 Inserir nota 3 : 1 Inserir nota 4 : 9 Inserir nota 5 : 7</p>
<u>Estado das variáveis</u>	<u>Simulação</u>
<p>j= 5</p>	<p>notas</p>  <p>0 1 2 3 4</p>

Exemplo 2 I

Modificar o programa anterior de forma tal que depois de ter inserido as 5 notas seja calculada a média.

Exemplo 2 II

```
int main(){
    int i, notas[5];
    double acum = 0, media;
    //Inserção de dados
    for (j = 0; j < 5; j++){
        cout << "Inserir nota"<< j+1;
        cin >> notas[j];
    }
    //Calculo da media
    for (j = 0; j < 5; j++)
        acum = acum + notas[j];
    media = acum / 5;
    cout<<"A media eh: " << media;
    return 0;
}
```

Exemplo 3 I

Construa um algoritmo que leia 300 números inteiros e imprima esses números na ordem inversa de entrada.

Exemplo 3 II

```
int main(){
    int i, num[300];
    //Inserção de dados
    for (i = 0; i < 300; i++){
        cout << "Inserir numero" << i+1;
        cin >> num[i];
    }
    //Visualização de dados
    for (i = 299; i >= 0; i--){
        cout << num[i];
    }
    return 0;
}
```


Exemplo 4

- ▶ Colocar os números de 1 a 5 em Vetor

Exemplo 4

- ▶ Colocar os números de 1 a 5 em Vetor

```
for(i=0; i<5; i++)  
    Vetor[i] = i+1;
```

Exemplo 4

- ▶ Colocar os números de 1 a 5 em Vetor

```
for(i=0; i<5; i++)  
    Vetor[i] = i+1;
```

- ▶ Colocar os números de 5 a 1 em Vetor

Exemplo 4

- ▶ Colocar os números de 1 a 5 em Vetor

```
for(i=0; i<5; i++)  
    Vetor[i] = i+1;
```

- ▶ Colocar os números de 5 a 1 em Vetor

```
for(i=0; i<5; i++)  
    Vetor[i] = 5-i;
```

Inicializando vetores I

- ▶ Quando declaramos um vetor, os seus elementos não são inicializados.
- ▶ Quando é declarado um vetor é possível atribuir valores iniciais.
- ▶ O valores iniciais são colocados entre chaves {}

```
int v[5] = {16, 36, 3, 8, 26};
```

Inicializando vetores II

- ▶ A quantidade de valores entre chaves não deve ser maior que o número de elementos
- ▶ A fim de facilitar a inicialização, C++ permite deixar o número de elementos em branco [] .
- ▶ Neste caso, o compilador vai supor que o tamanho do vetor é igual ao número de valores especificados entre chaves {}

```
int v[] = {16, 36, 3, 8, 26};
```

Uso de constantes em Vetores I

```
#define TAM_MAX 10
...
double VetReais[TAM_MAX];
// coloca 5,4,3,2,1 no vetor
for(i=0; i<TAM_MAX; i++)
    VetReais[i] = TAM_MAX - i;
```

```
// copia os dados de um vetor para outro
#define TAM_MAX 20
double VetReais[TAM_MAX], VetCopia[TAM_MAX];
for(i=0; i<TAM_MAX; i++)
    VetCopia[i] = VetReais[i];
```

Uso de constantes em Vetores I

```
const int TAM_MAX = 10;
...
double VetReais[TAM_MAX];
// coloca 5,4,3,2,1 no vetor
for(i=0; i<TAM_MAX; i++)
    VetReais[i] = TAM_MAX - i;
```

```
// copia os dados de um vetor para outro
const int TAM_MAX = 20;
...
double VetReais[TAM_MAX], VetCopia[TAM_MAX];
for(i=0; i<TAM_MAX; i++)
    VetCopia[i] = VetReais[i];
```


Busca I

Dada uma coleção de n elementos, pretende-se saber se um determinado elemento valor está presente nessa coleção. Para efeitos práticos, vamos supor que essa coleção é implementada como sendo um vetor de n elementos inteiros:
`vetor[0] .. vetor[n-1]`.

Pesquisa sequencial I

Uma solução possível é percorrer o vetor desde a primeira posição até a última. Para cada posição i , comparamos `vetor[i]` com valor.

- ▶ Se forem iguais dizemos que valor existe.
- ▶ Se chegarmos ao fim do vetor sem sucesso dizemos que valor não existe.

Pesquisa sequencial II

Passos:

1. Inicialização

```
i = 0;  
encontrado = 0; /*Falso*/
```

Pesquisa sequencial III

2. Pesquisa

```
while (i < TAMANHO && !encontrado) {  
    if (vetor[i] == valor) {  
        encontrado = 1; /*Verdadeiro*/  
    }  
    else {  
        i++;  
    }  
}
```

Pesquisa sequencial IV

3. Tratamento do resultado

```
if (encontrado) {  
    cout << "Valor " << vetor[i]  
        << "esta na posicao " << i;  
}  
else {  
    cout << "Valor nao encontrado \n";  
}
```

Pesquisa sequencial V

```
int main(){
    int num[10], valor, encontrado = 0, i;
    // inserir valores
    for (i = 0; i < 10; i++)
        cin >> num[i];
    cout << "Valor procurado";
    cin >> valor;
    // Busca
    for (i = 0; i < 10 && !encontrado; i++)
        if (num[i] == valor) encontrado = 1;
    if (encontrado)
        cout << "esta na posicao " << i-1;
    else
        cout << "Valor nao encontrado";
    return 0;}
```

FIM