

Programação



UFOP



INSTITUTO FEDERAL
MINAS GERAIS
Campus Ouro Preto



Vetores

Vetores

- Um vetor é um conjunto de variáveis
 - Do mesmo tipo;
 - Referenciadas pelo mesmo identificador (nome);
 - Cada variável é diferenciada das outras por um número, chamado de índice.

```
vector<int> nota[9];
```

Índice	0	1	2	3	4	5	6	7	8
Valor	6.5	4.3	7.8	9.8	10.0	9.8	8.7	10.0	10.0

Vetores

- Vetores são variáveis compostas homogêneas unidimensionais;
- Sintaxe

vector<tipo> identificador(dimensao);

- Em que:
 - **tipo**: é um tipo da linguagem C++;
 - **identificador**: é um identificador válido em linguagem C++;
 - **dimensao**: é o número de posições, um número inteiro, que determina quantos valores poderão ser armazenados no vetor.
- Note que é necessário incluir a biblioteca **<vector>**.

Vetores

```
#include <vector>
using namespace std;
int main()
{
    vector<int> numeros(50);
    vector<char> letras(10);
    vector<float> notas(14);
    vector<double> aproximacoes(999);

    return 0;
}
```

Vetores

- Somente um identificador é utilizado
 - A distinção entre as posições é feita pelo índice.
- O índice permite o acesso direto a uma determinada posição
 - Não é necessário percorrer todas as outras até chegar na desejada.

Vetores

```
int main()
{
    vector<int> numeros(50);

    numeros[4] = 1;
    numeros[10] = 2;
    numeros[49] = 3;

    return 0;
}
```

Vetores

Atenção!

- A primeira posição de um vetor sempre é a posição **zero**;
- A última posição é sempre **tamanho-1**;
- É possível tentar escrever/ler uma posição indevida no vetor
 - O compilador não faz esta verificação;
 - Causa erro de execução!


```
int main()
{
    vector<int> numeros(5); //o vetor possui 5 posicoes

    numeros[0] = 1; //primeira posicao
    numeros[1] = 2;
    numeros[2] = 3;
    numeros[3] = 4;
    numeros[4] = 5; //última posicao
    numeros[5] = 6; //erro!
    numeros[6] = 7; //erro!
    return 0;
}
```

Vetores

- As posições de um vetor podem ser utilizadas em expressões algébricas e lógicas
 - Cuidado para o tipo resultante da expressão não ser diferente do tipo do vetor.

```
int main()
{
    vector<int> numeros(5);

    numeros[0] += numeros[1];

    cout << numeros[3] << endl;

    numeros[1] = numeros[2] * numeros[3];

    if(numeros[3] >= numeros[4])
        numeros[3]++;
    return 0;
}
```

Vetores e Estruturas de Repetição

- Para percorrer ou preencher um vetor, utilizamos estruturas de repetição
 - Qualquer uma delas;
 - Porém, a instrução *for* é a mais utilizada, por causa do seu contador, que pode ser utilizado como índice.

```
#define TAMANH0 5
```

```
int main()  
{  
    vector<int> vetor(TAMANH0);  
  
    printf("Digite %d números\n", TAMANH0);  
  
    for(int i=0; i<TAMANH0; i++)  
        cin>>vetor[i];  
  
    for(int i=0; i<TAMANH0; i++)  
        cout<<vetor[i]<<endl;  
  
    return 0;  
}
```

Ordenação

- Frequentemente nos deparamos com a tarefa de ordenar valores numéricos armazenados em um vetor;
- Felizmente, temos um algoritmo de ordenação eficiente já implementado na biblioteca padrão do C++;
- Por padrão, o algoritmo ordena um vetor de modo crescente, mas podemos alterá-lo para ordenar de modo decrescente.

```
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

int main(){
    vector<int> vetor(TAMANHO);

    printf("Digite %d números\n", TAMANHO);
    for(int i=0; i<TAMANHO; i++)
        cin>>vetor[i];

    //ordena de forma crescente
    sort (vetor.begin(), vetor.end());
    return 0;
}
```

```
//função auxiliar
bool decrescente (int i,int j){
    return (i>j);
}

int main(){
    vector<int> vetor(TAMANHO);

    printf("Digite %d números\n", TAMANHO);
    for(int i=0; i<TAMANHO; i++)
        cin>>vetor[i];

    //ordena de forma decrescente
    sort (vetor.begin(), vetor.end(), decrescente);
    return 0;
}
```


Mandamentos do Uso de Vetores

1. Não alocarás dinamicamente;
2. Declararás o vetor com tamanho **maior** do que o limite máximo do problema;
3. Limparás o vetor antes de usar;
4. Não cobiçarás a memória do próximo.

Problemas Seleccionados

- ▣ <http://br.spoj.com/problems/PUSAPO11/>
- ▣ <http://br.spoj.com/problems/FLIPERAM/>
- ▣ <http://br.spoj.com/problems/MINADO12/>
- ▣ <http://br.spoj.com/problems/ELEICOES/>
- ▣ <http://br.spoj.com/problems/IMPEDIDO/>



Perguntas?