



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa



Projeto Olímpico de Programação

[C++] Vetores

Vetores

- Unidimensionais e bidimensionais;
- Seguem o mesmo padrão adotado em C.

Vetores

```
#include <iostream>

using namespace std;

int main() {
    int vetor[6] = {}, matriz[2][2] = {};

    for (int i = 0; i < 6; ++i)
        cout << vetor[i] << " ";

    cout << endl;

    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 2; ++j)
            cout << matriz[i][j] << " ";
        cout << endl;
    }

    return 0;
}
```



Alocação Dinâmica de Memória

Alocação Dinâmica de Memória

- Operador “new”:
 - Aloca um bloco de memória;
 - Retorna o endereço do bloco alocado, caso contrário (memória indisponível) retorna NULL.

Alocação Dinâmica de Memória

- **Exemplo:**

Definir uma estrutura “Ponto” contendo dois inteiros, alocar memória para armazenar um único ponto. Ao final o programa deverá liberar a memória.

```
#include <iostream>

using namespace std;

typedef struct{
    int x, y;
}Ponto;

int main(){
    Ponto *p;

    p = new Ponto;

    if (p != NULL){
        cout << "Memoria alocada." << endl;
        delete p;
        cout << "Memoria liberada." << endl;
    }else{
        cout << "Sem memoria";
    }

    return 0;
}
```



Alocação Dinâmica de Memória

- **Exemplo:**

Programa para obter do usuário a quantidade de números que ele deseja armazenar, alocar memória para essa quantidade e liberar a memória.

```
#include <iostream>

using namespace std;

int main(){
    int *pt, qtde;

    cin >> qtde;
    pt = new int[qtde];

    if (pt != NULL){
        cout << "Memoria alocada." << endl;
        delete[] pt;
        cout << "Memoria liberada." << endl;
    }else{
        cout << "Sem memoria";
    }

    return 0;
}
```



Alocação Dinâmica de Memória

- **Exemplo:**

Programa para obter um número, alocar memória para uma matriz quadrada de ordem “número”. Ao final o programa deverá liberar a memória.

```
#include <iostream>

using namespace std;

int main(){
    int **pt, linhas, colunas;

    cin >> linhas >> colunas;

    pt = new int*[linhas];

    if (pt != NULL){
        for (int i = 0; i < linhas; ++i)
            pt[i] = new int[colunas];
        cout << "Memoria alocada." << endl;
        delete[] pt;
        cout << "Memoria liberada." << endl;
    }

    return 0;
}
```

