

Aula 3: Material de consulta - Vetores

Introdução a vetores:

- **O que é vetores:** São estruturas de dados que permitem armazenar uma sequência de elementos do mesmo tipo em uma única variável. Eles podem ser usados para organizar e manipular coleções de dados de forma eficiente. Podem ser conhecidos como: *arrays*, *arranjos* ou *sequência de elementos*.

Tipos de vetores:

- **Estático:** São declarados com um tamanho fixo, que não pode ser alterado durante a execução do programa. Nesse exemplo é utilizado uma atribuição de 5 valores ao vetor, dessa forma ele não poderá sofrer alteração ao longo da compilação do programa.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int main()
5 {
6     int vetor [5];
7
8
9     return 0;
10 }
```

- **Dinâmico:** São declarados podendo sofrer alteração de tamanhos, durante o programa, ou seja, podem crescer ou diminuir de tamanho automaticamente conforme elementos são adicionados ou removidos. Nesse exemplo, podemos observar que o vetor é atribuído por “x” valores, dessa forma podendo alterar quando compilar o programa.

```
source code
1
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int main()
6 {
7     int vetor [x];
8
9     return 0;
10 }
```

Funções do vetor:

- **vet.push_back:** Insere um elemento ao final do vetor, (deve possuir o mesmo tipo de elementos do vetor).
- **vet.at:** Retorna o valor da posição, (deve ser inteiro).
- **vet.end e vet. begin:** Retorna um iterador apontando para a primeiro ou última posição do vetor, (respectivamente).
- **vet.size:** Retorna o tamanho do vetor, (sendo do mesmo tipo).
- **reverse (vet.begin(), vet.end()):** Faz a inversão do vetor, tanto do início para ao fim, como do final para o início.
- **cout(vet.begin(), vet.end(), val):** Retorna a quantidade de vezes que “val” está presente no vetor.

Resumo:

- Um vetor é precisamente utilizado para organizar, procurar, adicionar, podendo ser flexível pois seu tamanho pode crescer ou diminuir. Ele é muito usado em jogos, programas de cadastros e algoritmos.

Dicas:

- Tenha uma excelente base sobre os princípios da programação em c++ tendo como o principal: conceitos bases, estruturas condicionais, laços de repetição.
- Caso haja alguma necessidade, revise e refaça exercícios para melhor fixação das matérias passadas, para uma melhor compreensão sobre vetores.
- Exercite sempre, faça o uso de sua interpretação de textos e semânticas sobre os enunciados, entenda o as informações importantes antes de começar seu código.
- Procure usar as funções dos vetores para facilitar e avançar o seu conhecimento sobre a melhor forma de fazer o código de programação.

Obrigado, bons estudos !