

O objetivo desse exercício é praticar os conceitos de Templates em C++

Você deverá criar uma classe Vetor que funcione com qualquer um dos tipos primitivos do C++. Para isso, você deverá utilizar um Template. A sua classe deverá ter os seguintes métodos:

- Um método construtor que recebe um inteiro n como parâmetro e cria um vetor de tamanho máximo igual a n ;
- Um método construtor que recebe um outro vetor como parâmetro e faz uma cópia (construtor de cópia);
- Um método destrutor;
- Um método SetElemento que recebe um inteiro i e um elemento x e coloca o elemento na posição i do vetor.
- Um método GetElemento que recebe um inteiro i e retorna o elemento que está na posição i do vetor.
- Um método AdicionaElemento que recebe um elemento x e coloca o elemento na última posição do vetor
- Um método Imprime que imprime todos os elementos do Vetor com um espaço entre cada um deles.

Você deve ter atributos para tratar adequadamente o número de elementos e o tamanho máximo do seu vetor, mas pode considerar que as chamadas dos métodos sempre irão possuir índices válidos.

Importante: a definição e implementação da sua classe deve ser feita em um mesmo arquivo chamado vetor.hpp. Ou seja, não deve haver um .cpp separado. Para referência: <https://isocpp.org/wiki/faq/templates#templates-defn-vs-decl> . O arquivo main.cpp que utiliza a sua classe será fornecido.

Um exemplo de código que vai utilizar a sua classe segue abaixo:

```
Vetor<char> a(5);  
for(char i='a'; i<='e'; i++)  
    a.AdicionaElemento(i);  
  
Vetor<char> b = a;  
b.SetElemento(1,b.GetElemento(4));  
b.SetElemento(2,'i');  
b.SetElemento(3,'o');  
b.SetElemento(4,'u');  
  
a.Imprime();  
b.Imprime();
```

A saída desse código seria:

```
a b c d e  
a e i o u
```