

# Laboratório de Estrutura de Dados

## Exercício Prático 2

- 1) Considere o procedimento "swap" que aceita dois ponteiros e realiza a troca de endereços entre si:

```
void swap(void* left, void* right)
{
    void* temp = left;
    left = right;
    right = temp;
};
```

O método é chamado conforme o exemplo a seguir:

```
void* left = new int[0];
void* right = new int[0];

swap(left, right);
```

Após a execução, *left* deve apontar para o endereço anterior de *right* e vice-versa. No entanto, há um problema com o procedimento. Corrija o erro.

- 2) Desenvolva uma função em C++, dada uma assinatura qualquer de três dígitos (por exemplo: [1, 1, 1]) que calcule  $n$  elementos da sequência "tribonacci". A referida função "tribonacci" funciona de maneira parecida com a que calcula a sequência de Fibonacci. No entanto para o cálculo, "tribonacci" soma os 3 (três) últimos dígitos.

Se começarmos com a assinatura [1, 1, 1], será gerada a seguinte sequência: [1, 1, 1, 3, 5, 9, 17, 31, ...].

Observações:

- A assinatura sempre conterá 3 números;
- $n$  será sempre um número inteiro não negativo;
- Se  $n=0$  então retorne um vetor vazio;
- Todos os vetores devem ser declarados usando alocação dinâmica de memória;