

---

## *Exercícios de Template*

---

1) Utilizando templates, escreva uma função imprime que receba dois números e imprima a soma na tela. Considere o caso do template especializado para valores booleanos tal que a soma seja equivalente ao **OU** lógico.

2) Deduza a saída do código abaixo:

```
#include <iostream>

using namespace std;
template <typename T>
void fun(const T & x)
{
    static int contador = 0;
    cout << "x = " << x << " contador = " << contador << endl;
    ++contador;
return;
}
int main()
{
    fun<int> (1);
    cout << endl;
    fun<int>(1);
    cout << endl;
    fun<double>(1.1);
    cout << endl;
    return 0;
}
```

3) Estude o seguinte código:

```
#include <iostream>

using namespace std;
template <typename T>
T max(T x, T y)
{
    return (x > y)? x : y;
}

int main() {
    cout << max(3, 7) << endl;
    cout << max(3.0, 7.0) << endl;
    cout << max(3, 7.0) << endl;
    return 0;
}
```

Considerando o código acima:

- a) Corrija o erro produzido pelo terceiro cout sem alterar o conteúdo de main, para isto, altere a função parametrizada.
- b) Reescreva o código sem utilizar templates.

4) Considere a função abaixo que realiza a troca de conteúdo de duas variáveis:

```
void troca(int &a, int &b)
{
    int tmp = a;
    a = b;
    b = tmp;
}
```

Caso um usuário deseje utilizá-la para diversos tipos é necessário reescrevê-la para cada tipo. Implemente essa função utilizando template.