

# Métodos de Programação - Trabalho 2

Matrícula: 16/0019311

Nome: Victor André Gris Costa

## Teste Converter.Vazio

Esse teste foi criado com o objetivo de criar o caso mais simples de número romano que seria a string vazia, que neste design é converte-se em 0. Nesse teste pode-se ver que o ambiente de compilação estava bem montado e que o programa estava funcionando bem com o GTest. Passar nesse teste significava cobrir o caso mais simples de entrada.

## Teste Converter.AlgarismoI

Esse teste teve como objetivo testar o mecanismo de soma do algoritmo, demonstrando os casos em que só havia o algarismo I. Passar nesse teste demonstra a capacidade de somar algarismos romanos I.

## Teste Converter.AlgarismoV

Esse teste teve como objetivo testar a capacidade do algarismo de comparar algarismos novos diferentes de I. Ele apenas testa o número V, mas ajudou a criar um método que pode-se generalizar para as outras letras. Passar nesse teste significava uma capacidade de abranger algarismo V e I sem contar com a subtração.

## Teste Converter.CasoEspecialIV

Esse teste propôs o mecanismo básico de subtração existente nos números romanos. Passar nesse teste significa que o algoritmo é capaz de calcular quando há subtração (anterior menor que o próximo) com algarismos I e V.

## Teste Converter.NumerosComX

Nesse teste testou-se todos os números que só continham I, V e X. Passar nesse teste significaria que a generalização dos mecanismos de soma e subtração estavam indo na direção correta.

## Teste Converter.NumerosComL

Nesse teste verifica-se novamente que a generalização estava funcionando, testando a adição de outra letra a tabela. Passar nesse teste mostra que o programa está indo na direção de converter qualquer romano.

### Teste Converter.Todos

Aqui testa-se todos os números romanos até 3999. Nesse teste garantimos que o programa funciona para conversão de todos os números romanos necessários. Passar nesse teste mostra que o algoritmo está funcional, mas não necessariamente blindado contra entradas erradas.

### Teste Converter.NumerosLowerCase

Neste teste começamos a passar entradas com letras minúsculas. Passar nesse teste garante-se uma robustez maior ao programa, evitando que coisas bobas como case da letra causassem problemas.

### Teste Converter.Invalidos

Este teste bota-se a prova padrões impossíveis em números romanos, garantindo que o programa retorne -1 para esses casos. Passar nesse teste comprova que seu programa segue as especificações de retornar -1 para romanos inválidos.