

Universidade de Brasília
Departamento de Ciência da Computação
Disciplina: Métodos de Programação
Código da Disciplina: 201600

Métodos de Programação - 201600

Trabalho 2

O objetivo deste trabalho é utilizar o desenvolvimento orientado a testes (TDD) para resolver o problema de conversão de algarismos romanos para algarismos arábicos. O número em algarismos romanos poderá ser no máximo 3000. A função deverá pegar uma string de tamanho até 30 caracteres e retornar um inteiro correspondente ao número romano. Deverá retornar -1 se o número romano for inválido. Todas as strings correspondentes a números romanos válidos devem retornar o valor correto (ex. XXX deve retornar 30). Todas as strings correspondentes a números romanos inválidos devem retornar -1 (ex. XXXX, VV, VX, etc. devem retornar -1)

O desenvolvimento deverá ser feito passo a passo seguindo a metodologia TDD. A cada passo deve-se pensar qual é o objetivo do teste e o significado de passar ou não no teste.

Os resultados podem ser comparados com os números em:

<http://numeracaoromana.babuo.com/numeros-romanos-de-1-a-3000>

Ou qualquer outra fonte.

1) O programa deverá ser dividido em módulos e desenvolvido em C ou C++. Utilize o padrão de codificação dado em: <https://google.github.io/styleguide/cppguide.html>

O código deverá ser devidamente comentado facilitando o entendimento do mesmo.

2) Faça um documento dizendo quais testes você fez a cada passo e o que passar neste teste significa.

3) O desenvolvimento deverá ser feito utilizando um destes frameworks de teste:

3.1) gtest (<https://code.google.com/p/googletest/>)

3.2) catch (<https://github.com/philsquared/Catch/blob/master/docs/tutorial.md>)

4) Deverá ser entregue o histórico do desenvolvimento orientado a testes feito através do github (<https://github.com/>)

5) Instrumente o código usando o gcov. Usando o gcov.

(<http://gcc.gnu.org/onlinedocs/gcc/Gcov.html>). O makefile deve ser modificado de forma incluir as flags -ftest-coverage -fprofile-arcs. Depois de rodar o executável rode gcov nomearquivo e deverá ser gerado um arquivo .gcov com anotação.

O gcov é utilizado para saber qual percentual do código é coberto pelos testes. Neste caso os testes devem cobrir pelo menos 80% do código por módulo.

6) Faça a análise estática do programa utilizando o *cppcheck*, corrigindo os erros apontados pela ferramenta

Utilize **cppcheck --enable=warning** para verificar os avisos

7) Deve ser gerada uma documentação do código usando o programa DoxyGen (<http://www.stack.nl/~dimitri/doxygen/>): O programa inteiro terá de ser documentado usando DoxyGen.

Devem ser enviados para a tarefa no ead.unb.br um arquivo zip onde estão compactados todos os diretórios e arquivos necessários. O documento deve estar na raiz do diretório. Todos os arquivos devem ser enviados compactados em um único arquivo (.zip) e deve ser no formato `matricula_primeiro_nome` ex: `06_12345_Jose.zip`. Deve conter também um arquivo `leiametext` que diga como o programa deve ser compilado.

Data de entrega:

22/ 4 /17

Pela tarefa na página da disciplina no ead.unb.br