

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

Métodos de Programação - 201600

## **Trabalho 2**

Utilizando o trabalho 1:

a) Faça todos os testes de funções de forma automatizada utilizando o arcabouço de teste gtest (<https://code.google.com/p/googletest/>). Utilize o g++ como compilador.

b) Faça assertivas de entrada e de saída. Estas assertivas devem ser colocadas no código através do comando **assert** ou **ifs**. Comentários no código também devem especificar quais são estas assertivas

c) Utilize assertivas de entrada e de saída como parte da especificação (comentários antes das funções).

d) Faça assertivas como comentários de argumentação.

e) Utilize assertivas estruturais: elas definem a validade de uma coletânea de dados, ou estruturas de dados, e dos estados associados a estes dados.

f) Coloque nos comentários antes das funções quais são as assertivas do **contrato na especificação**. Diga o que deve ser esperado da função cliente em relação à entrada e o que deve ser garantido pela função servidora na saída

g) Instrumente o código 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 teste devem cobrir pelo menos 80% do código por módulo.**

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

**O programa deve ser feito orientado a testes como mostrado em sala de aula. Primeiro se desenvolve o teste e depois se cria o código para passar neste teste.**

**E importante rodar o gcov e cppcheck durante o processo de codificação, quando é mais fácil resolver eventuais problemas.**

Atualize a documentação colocando as informações sobre as assertivas de entrada e saída, assertivas como argumentação, contrato na especificação, etc. Deve ser gerado um documento (pdf ou odt) que diz como cada função foi testada. Para cada caso deve constar:

- 1) Nome da função, parâmetros e significado dos parâmetros. Especificação da função
- 2) Para cada um dos testes em cada função
  - 2.1) Nome de cada teste
  - 2.2) O que vai ser testado
  - 2.3) Qual deve a ser a entrada
  - 2.4) Qual deve ser a saída
  - 2.5) Qual é o critério para passar no teste
  - 2.6) Se a sua função efetivamente passou no teste ou não

O programa e o módulo devem ser feitos em C, colocados no makefile e depurados utilizando o GDB.

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.

Data de entrega:

1/ 10 /15

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