

WARM-UP

INFORMAÇÕES GERAIS

O objetivo desse trabalho é estimular vocês a iniciarem o Trabalho Final (TF) mais cedo possível, preparando o ambiente de projeto.

- Pontuação: Essa avaliação vai **valer 30% da minha parte da nota**.
- Entrega: 02/11/2021, 23:59 (**Hard deadline**)
- Trabalho pode ser realizado sozinho ou em dupla, sem possibilidade de uma configuração diferente.

ESPECIFICAÇÃO DO PROBLEMA

Vocês devem testar o programa *identifier*, usado como exemplo no livro do Maldonado. O programa é responsável por determinar se um identificador é válido ou não. Um identificador válido deve começar com uma letra e conter apenas letras e dígitos. Além disso, deve ter no mínimo 1 e no máximo 6 caracteres de comprimento. É importante observar que o programa contém um defeito.

RECURSOS UTILIZADOS

Vocês devem utilizar como base o projeto criado no GitHub, disponível em: <https://github.com/rafaelgaribotti/travis-identifier>. Este projeto utiliza como compilador o **gcc**, além do **make** para automatizar a compilação, e ainda usa o **Travis CI** como ferramenta de *Continuous Integration*. Vocês devem utilizar também o **gcov** como ferramenta de análise de cobertura de código, além das seguintes ferramentas voltadas a teste: **cppcheck**, **valgrind** e **sanitizer**.

Vocês também devem usar uma ferramenta para descrever os testes. Eu sugiro o uso do **Unity**, já incluído no repositório, mas este ponto fica de livre escolha caso alguns grupos queiram pesquisar e usar outras ferramentas. Neste sentido, deixo a critério do grupo decidir qual ferramenta utilizar para este propósito. Algumas alternativas mais conhecidas incluem: **gtest**, **cpptest**, **catch**. Entretanto existem dezenas de outras opções.

Os alunos **podem propor ferramentas adicionais** e isso vai ser altamente valorizado na avaliação.

ENTREGÁVEIS

- Um arquivo zip com o código fonte do repositório e o relatório, **ambos postados no Moodle ANTES do prazo**.
- Os grupos devem se identificar pelo [Fórum do Moodle](#).
- O repositório deve ser totalmente automatizado com [make](#) em termos de: compilação, execução do relatório de cobertura, e execução dos testes. A cada *commit* o Travis CI deve ser executado para verificar se os testes passaram.
- O relatório deve ser no formato PDF, e conter:
 - Uma tabela com as classes de equivalências e valores limites e os testes resultantes do uso destes 2 critérios.
 - Ao descrever os testes no relatório, especifiquem no seguinte formato:

Número do Teste	Nome do Teste	Casos de Teste
1	<nome do teste1>	[{entrada},{saida esperada}]
2	<nome do teste2>	[{entrada},{saida esperada}]
...
N	<nome do testeN>	[{entrada},{saida esperada}]

- Especifiquem separadamente os testes adicionais, por exemplo, testes incluídos para aumentar a cobertura de código. Neste caso, além de adotar o mesmo formato para descrever o teste, inclua também uma frase justificando a inclusão desse teste. Por exemplo, ele cobre que parte do código? Que caso relevante que vocês perceberam que é necessário, mas que os critérios de equivalência e valor limite não contemplavam?
- Cobertura: o **gcov** suporta cobertura de linhas e de *branch*, mas como o programa é muito pequeno, quero que também considerem *cobertura de caminho* e *cobertura de predicados*. Para a cobertura de predicados, usem a técnica que mostrei em aula para montar a tabela verdade e definir quais os testes devem ser incluídos. Inclua os resultados obtidos de cobertura no relatório.