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

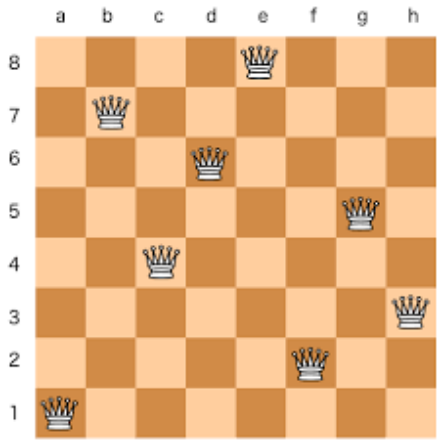
Métodos de Programação - 201600

Trabalho 1

O objetivo deste trabalho é utilizar o desenvolvimento orientado a testes (TDD) para **verificar** se um tabuleiro contém a solução para o problema das 8 rainhas.

Não é necessário encontrar uma solução. O objetivo é apenas verificar se a entrada é uma solução

Neste problema se verifica nenhuma das rainhas ataca as outras como na solução abaixo.



A entrada é uma sequência de caracteres.

00000100
01000000
00010000
00000010
00100000
00000001
00001000
10000000

Cada linha representa uma fileira do tabuleiro
‘0’ representa a posição vazia
‘1’ representa a rainha na posição

A função deverá retornar:
1 caso seja uma solução para o problema
0 caso não seja uma solução para o problema
-1 caso a entrada não seja válida. Ex. não é um tabuleiro 8x8, não tem 8 rainhas, etc.

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.

1) O programa deverá ser dividido em módulos e desenvolvido em C/C++ ou Python.
No caso de ser em C/C++, deverá haver um arquivo oito_rainhas.c (ou .cpp) e um arquivo oito_rainhas.h (ou .hpp). Deverá haver também um arquivo testa_oito_rainhas.c (ou .cpp) cujo objetivo é testar o funcionamento da biblioteca de teste se é uma solução para o problema das oitos rainhas.
No caso de ser em Python dever ser dividido em oito_rainhas.py (implementação) e testa_oito_rainhas.py (teste).

2) Se for em C/C++, utilize o padrão de codificação dado em:
https://google.github.io/styleguide/cppguide.html
quando ele não entrar em conflito com esta especificação. O código dever ser claro e bem comentado. O código deve ser verificado se está de acordo com o estilo usando o cpplint (https://github.com/cpplint/cpplint).

Se for em Python utilize o padrão de codificação dado em:
https://google.github.io/styleguide/pyguide.html
quando ele não entrar em conflito com esta especificação. O código deve ser verificado se está de acordo com o estilo usando o pylint
https://pypi.org/project/pylint/

Utilize o cpplint ou pylint desde o início da codificação pois é mais fácil adaptar o código no início.

3) Faça um documento (.txt ou .pdf) dizendo quais testes você fez a cada passo e o que passar neste teste significa.

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

C/C++
gtest (https://code.google.com/p/googletest/)
catch (https://github.com/philsquared/Catch/blob/master/docs/tutorial.md)

Python
Robot (https://robotframework.org/)
PyTest (https://docs.pytest.org/en/7.1.x/)

5) Deverá ser entregue o histórico do desenvolvimento orientado a testes feitos através do git (https://git-scm.com/docs/gittutorial)

git config --global user.name "Your Name Comes Here"
git config --global user.email you@yourdomain.example.com
git init
git add *
git commit –m “teste 1”
git log

Compactar o diretório “.git” ou equivalente enviando ele junto.

Devem ser enviados para a tarefa no aprender3.unb.br um arquivo zip onde estão compactados todos os diretórios e arquivos necessários. Todos os arquivos devem ser enviados compactados em um único arquivo (.zip) e deve ser no formato matricula_primeiro_nome.zip. ex: 06_12345_Jose.zip. Deve ser enviado um arquivo dizendo como o programa deve ser compilado e rodado.

Deve ser enviado o diretório “.git” compactado junto com o “.zip”

Data de entrega:

28/12 /22

Pela tarefa na página da disciplina no aprender3.unb.br