Assignment #3: N Queens

9/21/2023

10/10 Points

Attempt 1

Review Feedback
9/21/2023

Attempt 1 Score:
10/10

View Feedback

Anonymous Grading: **no**

Unlimited Attempts Allowed

Details

Implement a solution for the N-Queens problem using **stacks** (required), as shown in the lecture slides. Let the user define N (input) <= some Nmax (this should not affect the algorithm or its implementation details). N > 3. You do not need to implement your own stack. Your code should have sufficient comments so a reader can understand your logic. The output should be either the queens locations {rows, columns}, or (**bonus**) a graphical interface of the chess board with the queen locations (which can also be dynamic, during execution, showing the location changes).

Submit:

a single file, **X_Y_PA3.zip** or **X_Y_PA3.tar.gz**, where X is your first name (capital letters) and Y is your last name (capital letters). You get this file from compressing a folder named **X_Y_PA3** containing the following:

- Your code files (.cpp, .h, etc.)
- A Makefile file that contains all the commands needed to compile your code on Tesla@cs . All the code will be tested on Tesla@cs with its g++. Your code should be able to be compiled by executing *make*
- A README.txt file showing how the users should execute/run your program
- A screen recording of your compilation/execution

Bonus and penalties:

+2pts: graphical interface of chess board (see description)

-5pts: not using a stack according to the lecture slides

-2pts: N not user defined

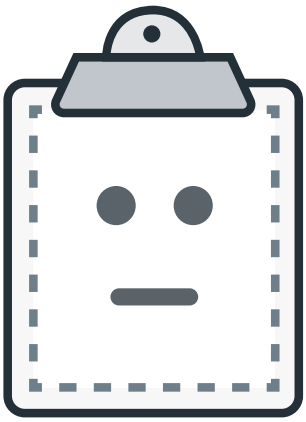
-1pts: insufficient comments in the code

-0.5pts: inadequate submission file name

-1pts: missing REAME

-2pts: missing Makefile

-1pts: missing screen recording



Preview Unavailable

PARMINDAR_SINGH_PA3.zip.zip

Download

(https://iu.instructure.com/files/162602287/download?download_frd=1&verifier=pleNMDX3XfXrXQaUKjyXOjgjXdnvf6wGU2Z4PavZ)

You are unable to submit to this assignment as your enrollment in this course has been concluded.