### Eight Queens Problem ( Due 23 Oct 2020 )

The Eight Queens Problem is a fairly old problem that has been well discussed and researched. The first published reference to this problem was in a German Chess magazine in 1848 by Max Bezzel. In 1850, Franz Nauck published all 92 solutions of the problem for an 8x8 board. S. Gunther in 1874 suggested a method for finding solutions by using determinants and J.W.L. Glaisher extended this method. E. Dijkstra published a detailed description of the solution of the problem as a depth-first backtracking algorithm.

The original statement of the problem ran as follows - how can we place eight queens on a regular chess board such that no queen can capture another. It turns out there is no unique solution but 92 possible solutions of which only 12 are distinct. The 12 distinct solutions can generate all other solutions through reflections and / or rotations. Here is a table that gives the size of the board, all possible solutions, and all distinct solutions.

|  |  |  |
| --- | --- | --- |
| **Size** | **All Solutions** | **Distinct Solutions** |
| 1 | 1 | 1 |
| 2 | 0 | 0 |
| 3 | 0 | 0 |
| 4 | 2 | 1 |
| 5 | 10 | 2 |
| 6 | 4 | 1 |
| 7 | 40 | 6 |
| 8 | 92 | 12 |
| 9 | 352 | 46 |
| 10 | 724 | 92 |
| 11 | 2680 | 341 |
| 12 | 14200 | 1787 |

This is a classic back-tracking problem and we have discussed the [solution and code](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/Chess.py)in class. The code that we worked on prints only one possible solution to the problem. There are several variations to this problem and their solutions.

Read the size of the board from a file [chess.in](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/chess.in). The number *n* will between 1 and 12 inclusive. Generate all possible solutions for a board of that size. Keep a count of the number of solutions and print the total number of solutions. For a board of size *8* your output should be *92*.

The file that you will be turning in will be called [Chess.py](https://www.cs.utexas.edu/users/mitra/csFall2020/cs313/assgn/Chess.py). We are looking for clean and structured design using the [standard coding conventions](https://www.python.org/dev/peps/pep-0008/)in Python. For this assignment you may modify any of the functions that we have provided and add any helper functions that you need. The file will have a header of the following form:

# File: Chess.py

# Description:

# Student Name:

# Student UT EID:

# Partner Name:

# Partner UT EID:

# Course Name: CS 313E

# Unique Number:

# Date Created:

# Date Last Modified:

You may change any of the functions or add helper functions as needed. To run this code on the command line on the Mac you will do

python3 Grid.py < grid.in

To run the code on the command line on a Windows machine you will do

python Grid.py < grid.in

If you are working with a partner, you will be submitting only one program but make sure that you have your partner's name and eid in your program. If you are working alone, then remove the two lines that has the partner's name and eid in the header.

Use the [Canvas](http://canvas.utexas.edu/)system to submit your **Chess.py** file. We should receive your work by 11 PM on Friday, 23 Oct 2020. There will be substantial penalties if you do not adhere to the guidelines. Remember Python is case sensitive. The name of your file must match exactly what we have specified.

* Your Python program should have the proper header.
* Your code must run before submission.
* You should be submitting your file through the web based *Canvas* program. We will not accept files e-mailed to us.

#### References

1. [Eight Queens Problem Article in Wolfram MathWorld.](http://mathworld.wolfram.com/QueensProblem.html)
2. [Article on Eight Queens Problem in Wikipedia](http://en.wikipedia.org/wiki/Eight_queens_puzzle) Watch the animation of the recursive solution.