CS 461
Spring 2018
Program 1
Due Sunday night, Feb 4.

Your first programming assignment uses unguided search to solve a game problem.

THE PROBLEM:
Consider a simple game consisting of a series of *n* tokens, played by 2 players. The players alternate turns. Each player must take at least one token, and can take no more than *n*/2 (fractions round down). The player who takes the last token loses.

Given *n* tokens, what is the correct number to take to optimize the player's chance of winning?

There is an obvious recursive strategy, but the sub-problems overlap and are not independent; thus you will probably want to use a memoization strategy of some kind.

Output: For all values of *n* from 1 to 500, state the probability of the player 'on the move' winning, the number of tokens they should select to maximize their chance of winning, and the likelihood of winning if that number is taken.

Your program output should be more than just a list of numbers; make it clear how many coins you're starting with, how many should be taken, and which player's chances of winning you're talking about.

Prepare a short report documenting your code and explaining how you implement the search, and handle the game logic. Your program may be in C++, Java, Python, or Clojure. Upload the source code and report to Blackboard by the deadline.