Alex Hamlin

Intro to AI: Project 1

**Data Structure:** I decided to use arrays to store my values for the cube. Once my values were stored in the arrays, I was able to lay them out side-by-side like a flattened cube. Using NumPy I was able to work with and manipulate the arrays much easier. My file is a standard Python3 file. I made it on Visual Studio 2015. I included the whole VS file if needed but I clearly named my python Source file if that is easier to run.

My randomizer takes a number of moves from the user. It will then multiply that number by up to 3. Since each time the face of a cube is turn, it can be turned up to 3 times without coming back to its original face, I just used a random number generator to pick 1-3 rotations per move. I believe that my code will work to randomize the whole cube however I ran into a wall while trying to return the new arrays from my randomizing functions. I could not get the arrays to correctly implement back into the functions to continue jumbling up the sides. Every time it goes to a new set of moves, it returns to the base state cube before randomizing it. So, in short, the randomizer only randomizes the face layer/top of the cube each time. However, it does this very well. My randomizer and data structure are in the same code. When it runs it just asks the user how many to jumble it.

I have decided that the heuristic for my cube in its current condition would be 1. Since my cube, unfortunately, only jumbles the top layer then it can never be more than 2 turns away from being solved. 66% of the time it will be within 1 turn of being solved. I will have a better heuristic when my cube works correctly.

Although my cube does not work correctly, I believe that I am a short fix away from it working as intended. I enjoyed trying to think how a cube would be laid out and the actually trying to get it into a working form. I look forward to trying to figure out how to solve the cube for the next iteration. I learned to use a new tool for Python (NumPy) which I can see being very useful in the future.