Project in Algorithmic Game Theory 2013. Hand in by July 4, 2013.

Consider the following symmetric 2-player game G in which every player has two available actions, 0 and 1.

1. If both players play 0 then their payoff is 0.
2. If both players play 1 then their payoff is 1.
3. If one player plays 0 and the other plays 1, then the payoff for the 0 player is $+2$ and the payoff for the 1 player is -2.

The game G is embedded in a competition as follows. There are $n > 2$ players in the competition numbered from 0 to $n-1$, and there are 100 rounds in the competition. There is a *coordinator* for the competition. All communication of players with the other players happens only through the coordinator. Every round contains n matches, where each match is a game of G. Every match has a *leader* and a *follower*. In match i within a round, the coordinator notifies player i that he is the current leader. The leader needs to reply with two pieces of information: one is a choice of another player that will be the follower in the match (hence a number $j \neq i$ in the range $0 \leq j \leq n-1$), and the other is the move to be played in the match (either 0 or 1). The coordinator then notifies the follower that he is invited to a match by player i (but the move of i is not revealed to the follower). The follower needs to reply with a move. The coordinator reports both to the leader and to the follower what the other player played, and hence they can compute their own payoffs. This ends match i, and the coordinator moves to match $i+1$ (modulo n), notifies player $i+1$ that he is the current leader, and so on.

Competition rules.

1. Every student who takes this course for credit needs to submit one agent. Students not taking the course for credit cannot participate in the competition.
2. n will be set to be the number of agents submitted. The mapping of agents to identities in the range $[0, n-1]$ will be done at random and kept secret (every agent will be informed only of n and its own number).
3. The agents need to run in "reasonable" time and space. (The tournament involves hundreds of matches, so spending one second per match is considered reasonable.)

4. The agents need to be written in Python and be compatible with the instructions provided by the grader.

5. *This is not a competition in programming.* If you find it difficult to learn Python or to program in it, you are free to use whatever help you can get.

6. For the purpose of reproducibility of results, the programs for the agents need to be deterministic. (Note: you may use randomness in the process of writing the program, for example, by including as part of the program a table of numbers previously produced at random.)

7. The grade in the project is composed of 80 points for submitting a program that works properly during the competition, plus bonus points which are computed as follows. One takes the total number of points accumulated by the agent in all rounds, and divides it by 10. For example, if an agent accumulated 100 points the final grade is 90, if an agent accumulated -100 points the final grade is 70, and if an agent accumulated 243 points the final grade is 104.3.

8. Students may submit agents either individually or in pairs. Each of the two students in a pair gets the whole grade for the project.

9. The results of the tournament will be kept confidential. Students will only be told their own grades, and not the grades of other students.

10. Nevertheless, some aggregate statistics regarding the tournament may be made public. These include the exact value of n, and rough estimates (not exact numbers) of how many matches of each type there were altogether (matches in which both players played 0, matches in which both players played 1, matches in which the leader played 0 and the follower played 1, matches in which the leader played 1 and the follower played 0).