
Algorithm 1 Computer vs. Computer Prisoner's Dilemma Game Simulation

```
1: procedure PRISONERDILEMMA(player1_choice, player2_choice)
2:   Define betrayal_reward, cooperation_reward, temptation_reward, punishment_reward
3:   if both players betray then
4:     return punishment_reward for both
5:   else if both players cooperate then
6:     return cooperation_reward for both
7:   else if player1 betrays and player2 cooperates then
8:     return temptation_reward for player1, 0 for player2
9:   else if player1 cooperates and player2 betrays then
10:    return 0 for player1, temptation_reward for player2
11:   end if
12: end procedure
13: procedure COMPUTERSTRATEGYRANDOM(opponent_last_move)
14:   return random choice between "cooperate" and "betray"
15: end procedure
16: procedure COMPUTERSTRATEGYALWAYS BETRAY(opponent_last_move)
17:   return "betray"
18: end procedure
19: procedure COMPUTERSTRATEGYALWAYS COOPERATE(opponent_last_move)
20:   return "cooperate"
21: end procedure
22: procedure COMPUTERSTRATEGYTITFORTAT(opponent_last_move)
23:   return opponent_last_move
24: end procedure
25: procedure COMPUTERSTRATEGYC(opponent_last_move)
26:   if random chance then
27:     return "betray"
28:   else
29:     return opponent_last_move
30:   end if
31: end procedure
32: procedure COMPUTERSTRATEGYD(opponent_last_move)
33:   if random chance then
34:     return "cooperate"
35:   else
36:     return opponent_last_move
37:   end if
38: end procedure
39: procedure COMPUTERSTRATEGYXB(opponent_last_move)
40:   if random chance with 70% bias then
41:     return "betray"
42:   else
43:     return "cooperate"
44:   end if
45: end procedure
46: procedure COMPUTERSTRATEGYXC1(opponent_last_move)
47:   if random chance with 70% bias then
48:     return "cooperate"
49:   else
50:     return "betray"
51:   end if
52: end procedure
53: procedure MAIN
```