Pseudocode for Repeated Prisoner's Dilemma Simulation (Computer vs Computer)

Repeated Prisoner's Dilemma Pseudocode (Computer vs Computer)

Algorithm 1 Main Function: Run Simulation

```
Initialize empty list data_list for storing game data (rounds and scores)
Input: Number of rounds to play, rounds
Define available strategies for computers
Print available strategies for selection
Input: User selects computer_strategy1 for Computer 1
Input: User selects computer_strategy2 for Computer 2
Initialize scores for both computers: player1_score and player2_score to 0
Set initial choices for both computers to "cooperate"
for each round_num from 1 to rounds do
    {Main game loop over number of rounds}
    computer_choice1 $\leftarrow$ computer_strategy1(computer_choice2) {Computer 1 decides based on strategy and opponent's last move}
    computer_choice2 $\leftarrow$ computer_strategy2(computer_choice1) {Computer 2 responds similarly}
    reward_player1, reward_player2 $\leftarrow$ Call prisoner_dilemma with computer_choice1, computer_choice2
    Update cumulative scores: player1_score += reward_player1, player2_score += reward_player2
    Append current round data to data_list using add_data
end for
Print "Game Over!" and final scores for both computers
Export the data to CSV and Excel using excel(data_list)
```

Algorithm 2 Prisoner’s Dilemma Function

```
Input: player1_choice, player2_choice
Define rewards: betrayal_reward $\leftarrow$ 5, cooperation_reward $\leftarrow$ 3, temptation_reward $\leftarrow$ 8, punishment_reward $\leftarrow$ 1
if player1_choice is ”betray” AND player2_choice is ”betray” then
    Return punishment_reward for both
else if player1_choice is ”cooperate” AND player2_choice is ”cooperate” then
    Return cooperation_reward for both
else if player1_choice is ”betray” AND player2_choice is ”cooperate” then
    Return temptation_reward for Player 1, and 0 for Player 2
else if player1_choice is ”cooperate” AND player2_choice is ”betray” then
    Return 0 for Player 1, and temptation_reward for Player 2
end if
```

Algorithm 3 Add Data Function

```
Input: data_list, round_num, player1_score, player2_score
Append current round data (round number and player scores) to data_list
```

Algorithm 4 Export Data to CSV/Excel

```
Input: data_list
Write data_list to CSV file
Write data_list to Excel file using pandas
Print confirmation of file creation
```

Algorithm 5 Random Strategy

```
Input: x (opponent’s last choice)
Return ”cooperate” or ”betray” randomly
```

Algorithm 6 Always Betray Strategy

```
Input: x (opponent’s last choice)
Always return ”betray”
```

Algorithm 7 Always Cooperate Strategy

```
Input: x (opponent’s last choice)
Always return ”cooperate”
```

Algorithm 8 Tit-for-Tat Strategy

```
Input: x (opponent’s last choice)
Return opponent’s last action
```

Algorithm 9 Strategy C (50% chance to betray or mimic)

```
Input: x (opponent’s last choice)
Return ”betray” with 50% probability, or mimic opponent’s last action
```

Algorithm 10 Strategy D (50% chance to cooperate or mimic)

```
Input: x (opponent’s last choice)
Return ”cooperate” with 50% probability, or mimic opponent’s last action
```

Algorithm 11 Strategy Xb (Bias towards betrayal, 70%)

```
Input: x (opponent’s last choice)
Return ”betray” with 70% probability, or ”cooperate”
```

Algorithm 12 Strategy Xc (Bias towards cooperation, 70%)

```
Input: x (opponent’s last choice)
Return ”cooperate” with 70% probability, or ”betray”
```