Prisoner's Dilemma Tournament

Objective:

Design a Python program that competes in a 100-round Prisoner's Dilemma tournament against programs submitted by other students. The program with the most wins across all matches is the champion.

Rules:

- **Choices:** In each round, your program must return either:
  - "we good" (Cooperate)
  - "throwing hands" (Defect)
- **Payoff:**
  - Both "we good": 1 point each
  - One "we good", one "throwing hands": 2 points for defector, 0 for cooperator
  - Both "throwing hands": 0 points each
- **Rounds:** 100 consecutive rounds per match
- **Information:** Your program receives the full history of both its own and its opponent's choices from previous rounds.

Technical Requirements:

- **Language:** Python 3
- **Libraries:** No external libraries allowed (use standard Python features only)
- **Submission:** Submit your Python function to the designated competition page.

Example Program:

```python
def exampleSubmission(yourPastPlays, yourOpsPastPlays):
    Y = "we good"
    N = "throwing hands"
    if len(yourPastPlays) == 0 or yourOpsPastPlays[-1] != N:
        return Y  # Cooperate initially or if opponent didn't defect
    return N # Defect if opponent defected last round (tit-for-tat)
```

Evaluation:

Results will be posted immediately on the competition page, showing how your program performed against every other submission.

Strategy Tips:

- The Prisoner's Dilemma explores concepts of trust, cooperation, and betrayal.
- Consider different strategies, such as tit-for-tat, always cooperating, always defecting, random choice, or more complex approaches based on the history of plays.

Get Creative! This is your chance to outsmart your classmates and design a winning strategy. Good luck!