A Tricky Problem

The genre of trick-taking card games encompasses many popular games, such as Bridge, Hearts, Euchre, and Briscola. At their core, these games revolve around each player having some number of cards (which may be dealt all at once, or as the game progresses), and a number of rounds (or “tricks”) where each player plays a card. Based on these cards, and specially on how they each compare to each other, players receive a number of points.

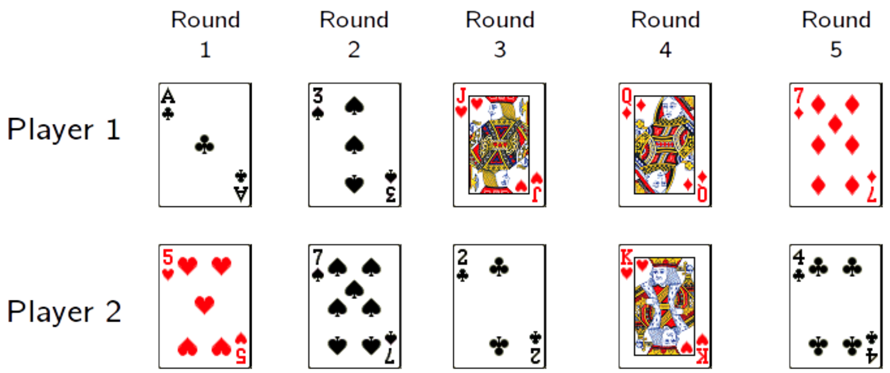
You will write a program to determine the winner of a trick-taking card, defined as follows:

- There are only two players.
- Each player is given N cards at the start of the game.
- The possible cards, from least valuable to most valuable, are:

  2, 3, 4, 5, 6, 7, 8, 9, J, Q, K, A

  Note that there is no 10 card.
- The suit of the card (hearts, diamonds, spades, clubs) is irrelevant.
- In each round, each player plays a single card (consequently, a single game is composed of N rounds). The player with the highest card in that round wins a single point. If both players play a card of equal value, no points are given in that round.
- At the end of the game, the player with the most points wins. If both players have the same number of points, the result of the game is a tie.

For example, consider the following game:



	Round 1	Round 2	Round 3	Round 4	Round 5
	Ace beats 5 Player 1 wins	7 beats 3 Player 2 wins	Jack beats 2 Player 1 wins	King beats Queen Player 2 wins	7 beats 4 Player 1 wins
Player 1 Score	1	1	2	2	3
Player 2 Score	0	1	1	2	2

At the end of this game, Player 1 will be the winner.

Your task is to implement the function `tricky`, which takes a list of cards for player 1 and a list of cards for player 2. It returns the string "PLAYER 1 WINS", if player 1 wins more rounds, the string "PLAYER 2 WINS", if player 2 wins more rounds, and the string "TIE" if the players win the same number of rounds.

Here are some sample games and the expected result:

Player 1	Player 2	Sample Output
A 3 J Q 7	5 7 2 K 4	PLAYER 1 WINS
A 3 J Q	5 7 2 K	TIE
2 2 2	A A A	PLAYER 2 WINS
A 3 2 Q 7	5 7 2 K 4	TIE

Here is the header file for this task:

```
#ifndef _Problem3_
#define _Problem3_
#include <vector>
#include <string>

/**
@class Problem
*/
class Problem3 {
public:

    /**
    Play a game of Tricky as described in the problem statement.

    @param player1  Vector of strings representing the moves of player 1
    @param player2  Vector of strings representing the moves of player 2

    @return the winner of the game, and "TIE" if there's no winner
    **/
    static std::string tricky(std::string& player1, std::string& player2);
};

#endif
```

And here is the skeleton code for this task:

```
#include "Problem3.h"
#include <string>
#include <iostream>

/**
    Play a game of Tricky as described in the problem statement.

    @param player1  Vector of strings representing the moves of player 1
    @param player2  Vector of strings representing the moves of player 2

    @return the winner of the game, and "TIE" if there's no winner
**/
std::string Problem3::tricky(std::string& player1, std::string& player2) {
    // YOUR CODE HERE
    return "";
}
```

When you take the placement exam, you will be expected to copy the the code for the header file and the skeleton code into specific files and then complete the function. For the practice problems, we have provided files `Problem3.h` and `Problem3.cpp` for your convenience.