Challenge: “World Cup Fever”
Level: Non-beginner

Description:

The US women’s national team recently won the 2019 Women’s World Cup. Other nations around the world have decided to work on their passing skills to improve their chances in the next World Cup. One particular country has decided to develop a computer models to simulate efficient passing among teammates in different game scenarios, and you have been chosen to develop the algorithm for this model.

The coaches will develop a series of game scenarios and use them as input for the passing simulation model, which should process that input and determine the minimum number of passes needed to get the ball from one specified player to another specified player. For the purposes of this simulation, the players are assumed to stay in the same position while the sequence of passes is performed.

In the simulation, Player P1 can pass the ball directly to Player P2 if:

- they are on the same team
- no other player (from either team) is in between the two players

For instance in the following scenario:

- There is a line formed from P1 to P2 to P3, in that order
- No other players are in between P1, P2, and P3

Then:

- If P1 and P2 are on the same team, they can pass
- If P2 and P3 are on the same team, they can pass
- Even if P1 and P3 are on the same team, they can not pass, since P2 is between them

(Please see the note on diagonal passes below in the Sample Solution portion.)

Example input set:

A string representing the setup of the players on the field, such that:

- The first token will be a number *n*, which represents the number of players on each team
- The next *n* tokens will represent the players of team 1
- The next *n* tokens will represent the players of team 2
- Player tokens will be formatted `R,C` – that is, a number representing the row, followed by a comma, followed by a number representing the column, which together represent the player’s position in a grid
- Players from the first team will always be passing the ball
- The ball will always start with the first player listed
- The ball should always be passed to Player *n* – that is, the last player listed for the first team.

```
3 0,3 0,1 3,3 1,3 2,3 3,2
n  P1   P2   P3   P4   P5   P6
```

(for informational purposes only -- not part of input set)
Players from team 1 in normal text; team 2 in ***bold italic***

Sample solution:

Creating a grid and marking the players on it would look like this:

	C0	C1	C2	C3
R0		P2		P1
R1				***P4***
R2				***P5***
R3			***P6***	P3

The ball (b) always starts with the first player listed, P1:

	C0	C1	C2	C3
R0		P2		P1 (b)
R1				***P4***
R2				***P5***
R3			***P6***	P3

The goal of team 1 is to get the ball to the *nth* player listed, which is P3. However P1 cannot pass to P3 due to ***P4*** and ***P5*** being between them. P1 can however pass the ball to teammate P2:

	C0	C1	C2	C3
R0		P2 (b)		P1
R1				***P4***
R2				***P5***
R3			***P6***	P3

There are no players from team 2 between P2 and P3, so P2 can complete the scenario and pass directly to P3:

	C0	C1	C2	C3
R0		P2		P1
R1				***P4***
R2				***P5***
R3			***P6***	P3 (b)

Note on diagonal passes:

Diagonal passes are allowed **except** under the condition that an opposing player is **exactly** on the path the ball would travel between two players. For instance, consider the following two scenarios:

	C0	C1	C2
R0	P1		
R1		***P3***	
R2			
R3			P2

	C0	C1	C2	C3
R0	P1			
R1				
R2			***P3***	
R3				P2

In the scenario on the left, ***P3*** is not *exactly* on the path between P1 and P2, so a pass between P1 and P2 is allowed.

In the scenario on the right, ***P3*** is *exactly* on the path between P1 and P2, so a pass between P1 and P2 is not allowed.
In mathematical terms, if the position R2,C2 of P2 can be described in terms of the position R1,C1 of P1 such that R1+r=R2 and C1+c=C2 *and* r = c, then a diagonal pass is not allowed if ***P3*** is on the path defined by R1+1..r,C1+1..c.

In the scenario on the left, the position of P2 defined in terms of the position of P1 is R1+2,C1+3. In this scenario, r != c, so ***P3*** cannot be on the path between P1 and P2, so a diagonal pass is allowed.

In the scenario on the right, the position of P2 defined in terms of the position of P1 is R1+3,C1+3. In this scenario, r = c, so we must check if ***P3*** is on the path between P1 and P2. It is easy enough to check both possibilities, namely: R1+1,C1+1 and R1+2,C1+2. Since P1 is at position 0,0 we can substitute 0 for both R1 and C1: 0+1,0+1 and 0+2,0+2 and we can see that we must check positions 1,1 and 2,2 for players on team 2. We do find ***P3*** at position 2,2 so a diagonal pass between P1 and P2 is not allowed.

PASS/FAIL:

The solution will be considered PASSED if it returns an output which:

- **Is properly formatted**
  The output will be a list of players who touched the ball such that:
  - All players are referred to as P#, where # is a valid number
  - The list starts with P1
  - The list ends with P*n* where *n* is the number of players on each team from the input set
  - There no spaces or other characters in the output

  The output from the sample input given above would be:

  ```
  P1P2P3
  ```

  If multiple paths are found, they should be submitted with a space between them. For example:

  ```
  P1P2P3P5  P1P4P2P3P5
  ```

  (The submission of multiple valid paths will be used in tiebreaker scenarios. See the Tiebreakers section below for more information. If multiple paths are submitted, all paths must be valid.)
- **Contains only valid players**
  Only players P1..*n* should be referred to, where *n* is the number of players on each team from the input set.
- **Contains only valid passes**
  Each pair of players next to each other in the output must be allowed to pass the ball between them; they must not have a player from team 2 between them, as defined above.
- **Does not use any player more than once**
  No player may handle the ball more than one time in any passing sequence. Each player may be passed the ball a maximum of one time.
- **Is unique**
  Multiple identical solutions for a single input set will not be considered valid.

Any solution which returns output that does not meet these criteria will be considered FAILED.

Tiebreakers:

The code you submit will be called several times, with a different input set each time. If more than one team outputs a correct solution to at least one input set, the following tiebreaker scenarios will be used to determine the ordering of teams for this challenge, in the order given below:

- Most input sets solved successfully, highest number winning the tiebreaker. Each input set will be weighted equally for this tiebreaker (e.g. no extra points for solving a “more difficult” input set).
- Total number of valid solutions found, summed across all input sets, highest number winning the tiebreaker. If Team A finds one valid solution for input set 1, two valid solutions for input set 2, and two valid solutions for input set 3, their total number of valid solutions is 5. Team B must find 6 or more valid solutions to win the tiebreaker.
- Submission time, earliest submission winning the tiebreaker.