

TwoPlayerGame FRQ

SHOW ALL YOUR WORK. REMEMBER THAT PROGRAM SEGMENTS ARE TO BE WRITTEN IN JAVA.

Assume that the classes listed in the Java Quick Reference have been imported where appropriate.

Unless otherwise noted in the question, assume that parameters in method calls are not *null* and that methods are called only when their preconditions are satisfied.

In writing solutions for each question, you may use any of the accessible methods that are listed in classes defined in that question. Writing significant amounts of code that can be replaced by a call to one of these methods will not receive full credit.

This question involves a simulation of a two-player game. In the game, two simulated players each start out with an equal number of coins. In each round, each player chooses to spend either 1, 2, or 3 coins. Coins are then awarded to each player according to the following rules.

Same rule: If both players spend the same number of coins, player 2 gains 1 coin.

Off-by-one rule: If the players do not spend the same number of coins and the positive difference between the number of coins spent by the two players is 1, player 2 is awarded 1 coin.

Off-by-two rule: If the players do not spend the same number of coins and the positive difference between the number of coins spent by the two players is 2, player 1 is awarded 2 coins.

The following is an example of a game played with a starting value of 10 coins and a game length of 5 rounds.

Round Number	Player 1 Coin Count at Beginning of Round	Player 2 Coin Count at Beginning of Round	Player 1 Spends	Player 2 Spends	Outcome	Player 1 Coin Count at End of Round	Player 2 Coin Count at End of Round
1	10	10	2	1	Off-by-one, player 2 gains 1 coin	$10 - 2 = 8$	$10 - 1 + 1 = 10$
2	8	10	2	2	Same, player 2 gains 1 coin	$8 - 2 = 6$	$10 - 2 + 1 = 9$
3	6	9	1	3	Off-by-two, player 1 gains 2 coins	$6 - 1 + 2 = 7$	$9 - 3 = 6$
4	7	6	2	2	Same, player 2 gains 1 coin	$7 - 2 = 5$	$6 - 2 + 1 = 5$
5	5	5	3	1	Off-by-two, player 1 gains 2 coins	$5 - 3 + 2 = 4$	$5 - 1 = 4$
End of Game	4	4			Tie game		

The game ends when the specified number of rounds have been played or when a player's coin count is less than 3 at the end of a round.

The `CoinGame` class is shown below. You will write two methods in the `CoinGame` class.

```
public class CoinGame

{
    private int startingCoins; // starting number of coins
    private int maxRounds; // maximum number of rounds played

    public CoinGame(int s, int r)

    {
        startingCoins = s;
        maxRounds = r;
    }

    /** Returns the number of coins (1, 2, or 3) that player 1 will spend.
    */
    public int getPlayer1Move()
    { /* implementation not shown. */ }

    /** Returns the number of coins (1, 2, or 3) that player 2 will spend, as
    described in part (a).
    */
    public int getPlayer2Move(int round)
    {
        /* to be implemented in part (a) */
    }

    /** Plays a simulated game between two players, as described in part (b).
    */
    public void playGame()
    {
        /* to be implemented in part (b) */
    }
}
```

In the simulation, player 2 will always play according to the same strategy. The number of coins player 2 spends is based on what round it is, as described below.

(a) You will write method `getPlayer2Move()`, which returns the number of coins that player 2 will spend in a given round of the game. In the first round of the game, the parameter `round` has the value 1, in the second round of the game, it has the value 2, and so on. The method returns 1, 2, or 3 based on the following rules.

If round is divisible by 3, then return 3.

If round is not divisible by 3 but is divisible by 2, then return 2.

If round is not divisible by 3 and is not divisible by 2, then return 1.

Complete method `getPlayer2Move()` below by assigning the correct value to result to be returned.

```
/** Returns the number of coins (1, 2, or 3) that player 2 will spend, as
described in part (a).
*/
public int getPlayer2Move(int round)
{
    int result;
    return result;
}
```

Write the method `playGame()`, which simulates a game between player 1 and player 2, based on the rules and example shown at the beginning of the question. Both player 1 and player 2 start the game with `startingCoins` coins. Computer player 1 spends 1, 2, or 3 coins based on the value returned by the method `getPlayer1Move()`. Computer player 2 spends 1, 2, or 3 coins based on the value returned by the method `getPlayer2Move()`.

The game ends when `maxRounds` rounds have been played or when a player's coin count is less than 3 at the end of a round.

At the end of the game, the winner is determined according to the following rules.

If both players have the same number of coins at the end of the game, the method prints "`tie game`".

If player 1 has more coins than player 2, the method prints "`player 1 wins`".

If player 2 has more coins than player 1, the method prints "`player 2 wins`".

(b) Assume that `getPlayer2Move()` works as specified, regardless of what you wrote in part (a). You must use `getPlayer1Move()` and `getPlayer2Move()` appropriately to receive full credit.

Complete method `playGame()` below.

```
/** Plays a simulated game between two players, as described in part (b).
*/
public void playGame()
```