```
//
//  Connect4.hpp
//  FreeRangeInterviewTest
//
//  Created by Stewart Bracken on 2/6/14.
//  Copyright (c) 2014 Stewart Bracken. All rights reserved.
//

#ifndef FreeRangeInterviewTest_Connect4Evaulator_hpp
#define FreeRangeInterviewTest_Connect4Evaulator_hpp

#include <vector>

typedef std::vector<char> conn4grid;

//Name of the game!
const int connect = 4;

const char red ='R',
           black = 'B',
           empty = '.';

class Connect4Evaulator {
    const int GRID_SIZE;

//****** PRIVATE METHODS *******//
private:
    int get_index (int x, int y, int _width);

    // Checks a tile against the next 3 using vx/vy as the direction.
    // Returns true if it's found a connect 4.
    bool has_connection4 (const conn4grid& connect4grid, char curr, int curr_x, int
curr_y, int vx, int vy, int width);

    //PRE: out_grid is empty
    void transpose (const conn4grid& grid, conn4grid& out_grid, int& width, int&
height);

    // Mirror flip rows
    void exchange_rows (conn4grid& grid, int width, int height);

    //Mirror flip columns
    void exchange_columns (conn4grid& grid, int width, int height);

    // Push all non-empty spaces downwards (increasing y)
```

```
    void apply_gravity (conn4grid& grid, int width, int height);


public:

    Connect4Evaulator(int grid_size = 42):GRID_SIZE(grid_size){}


//****** RETURN STATES *******//
    enum { RED_WIN, RED_LOSE, DRAW, UNFINISHED, NEITHER, LEFT, RIGHT, ERROR };


//****** PUBLIC METHODS *******//
    void print_grid (const conn4grid& grid, int width, int height);

    // Returns RED_WIN, RED_LOSE, DRAW, UNFINISHED, or ERROR
    int evaluate_conn4_state(const conn4grid& connect4grid, int width = 7, int
height = 6 );

    // Returns LEFT, RIGHT, NEITHER, or ERROR
    int evaluate_rolled_conn4_state (const conn4grid& original_grid, int width = 7,
int height = 6);


};


#endif
```