## Checkered Future

Due Sept 19th, Oct 1st
Purpose : classes, recursion, 2 dimensional Arrays, test plan, machine grading).
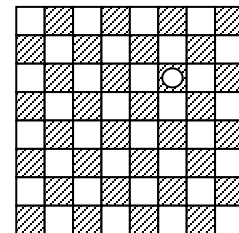
Universal Merriment Design (UMD) is building an electronic checkers game. They have hired you to write a small part of the game. Given the position on the board, you need to compute the maximum number of jumps of one specific king checker in one turn. Checkers is played on a board made up of 8 rows and 8 columns of alternating black and white squares. A king making a capturing move (a jump) leaps over one of the opponent's pieces, landing in a straight diagonal line on the other side and the opponents' piece is removed from the board. If the landing square is occupied, that jump is not legal. Only one piece may be captured in a single jump (and removed from the board); however, multiple jumps are allowed on a single turn
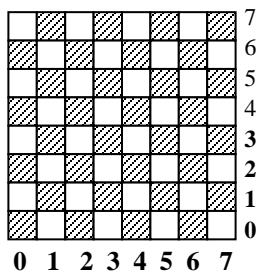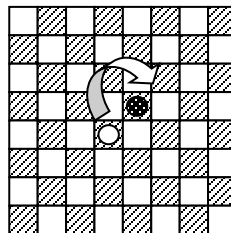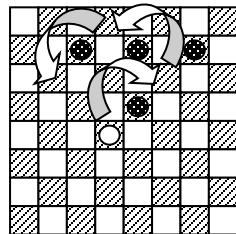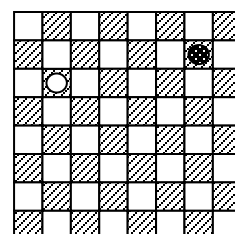

Checker board   **before making a jump**   **after making a jump**

before a triple jump   **after a triple jump**

Input:
The input will consist of multiple test cases. The first line of input will be an integer n, indicating the number of test cases. Each test case will begin with 2 integers (m and n) on 1 line , separated by a space, specifying the number of your pieces and the number of your opponent's pieces. The next m lines will contain the row ($0 \le r \le 7$) and column ($0 \le c \le 7$) information for each of your pieces separated by a space. The last n lines will contain the row ($0 \le r \le 7$) and column ($0 \le c \le 7$) information for each of your opponents pieces separated by a space.

Output:
Each test case will result in 1 line of output with the phrase, "*the number of jumps is x*", where x is the number of jumps your first piece (a king) can make legally.

Sample Input (from the single and triple jumps illustrated above)
```
2
1  1
```

```
3 3
4 4
1 4
3 3
4 4
6 6
6 2
6 4
```

Sample Output
```
the number of jumps is 1
the number of jumps is 3
```

If your program has no class, it will get a lower grade
If your program has global variables, it will get a lower grade
A function should fit on the screen. It should not be 1 line long.
You should write a routine to print out your board to help debug it.
Your description is written to someone who is your level but has not seen the problem
After reading your description, the user should be able to generate valid input to your
program and predict the output.

How the program will be graded

What _ indicates program/other is memo	Pts	Due Sept 19th	Due Oct 1st
External Documentation	**1**		
Your Name	1		X
Analysis	**15**		
Test Plan (a table with 4 columns Reason for the test, input, expected output, actual output) (4 test cases, each with a unique reason and NOT using the sample input)	15		X
Program Listing Style	**34**		
Your Name	1	X	X
Description of the problem (in your own words)	1	X	X
Style	18		X
Pre/Post conditions for functions	14		X
Functionality (minimum functions you should have)	**50**		X
`void addChecker(char who, int r, int c)` `// puts who at position (r,c)`	5	X	X
`void print() // prints the board`	10	X	X
`int main()`	10	X	X

`int numJumps(int row, int col)//`
`returns the number of jumps a checker`
`can make starting at (row, col)`
	20		X

`void reset( ); //clears the board`	5		X

Required Data Structure
You can add more to the data structure but the minimum should be :

```
Class Board
{
   public:
      Board(); //creates an empty board
      void reset( ); //clears the board
      void addChecker(char who, int row, int col);
      int numJumps(int row, int col); // num of jumps possible from row, col
      void print() ; // print the board for debugging only
   private:
     char board[8][8];
};
```