

Background:

For this project you will implement a program that plays Tic-Tac-Toe. Your program will be required to read the player's move from an input file, check the legality of the moves, output the, possibly, changed game board, check for a possible winner and determine if the game has ended.

Details:

Here is a sample input file, named TTTmoves.txt, for the program. The first line is a title line that can be skipped. The second line specifies the player's names and indicates which player is 'X' and 'O'. There will be a tab after the first player's name and the second player's label. The third line is blank. Each of the remaining lines will contain one move per line. A move consists of the player's label, ('X', 'O') followed by a single tab and the number of the square in which the player has selected to place their label.

```
Tic-Tac-Toe Game Data
X: Dave McPherson   O: Guy Rhodes

X      6
O      9
X      7
O      4
X      1
O      3
X      5
O      2
X      8
```

It may be assumed that each line contains the correct type of data and the correct number of values. Thus it is guaranteed the second line will contain the "X:" and "O:" labels and that each will be followed by two strings giving the players' first and last name. Each move line will contain a character followed by an integer. A game usually begins with a move by "X", but your program should be able to handle either player starting first. The game board will be numbered as follows:

1	2	3
4	5	6
7	8	9

The program must determine if a move is legal. Illegal moves are defined as follows:

1. A move by a player other than 'X' or 'O'
2. A move to an invalid square
3. A move by the player who moved last.
4. A move to a square that already contains a mark.

Illegal moves must be noted in the output file by echoing the invalid move followed by a brief error message. For the possible illegal moves listed above use the error messages given in the examples below:

```
Y    5    illegal player
X    0    illegal square
X    6    double move
O    1    marked square
```

The possible illegal moves must be checked for in the exact order given above. If a move contains more than one error, (e.g. Z0), only the first needs to be reported, (e.g. Z 0 illegal player).

After each move the program must determine if one player has won the game, (i.e. has three marks in the same row, column, or diagonal). Additionally, the program must determine if the game is a draw, (i.e. nine valid moves have been made and there is no winner). This is termed a “cat” game in tic-tac-toe. The program must halt as soon as one player wins or a “cat” game has occurred.

Below is a partial sample output file, named TTTgame.txt, which corresponds to the input data given above:

```
Tic-Tac-Toe game: X: Dave McPherson      O: Guy Rhodes

Move 1:   X    6
          -|-|-
          -+--+
          -|-|X
          -+--+
          -|-|-
          ~~~~~~

Move 2:   O    9
          -|-|-
          -+--+
          -|-|X
          -+--+
          -|-|O
          ~~~~~~

Move 3:   X    7
          -|-|-
          -+--+
          -|-|X
          -+--+
          X|-|O
          ~~~~~~
```

```

Move 4:   O    4
          -|-|-
          -+--+
          O|-|X
          -+--+
          X|-|O
~~~~~
Move 5:   X    1
          X|-|-
          -+--+
          O|-|X
          -+--+
          X|-|O
~~~~~
Move 6:   O    3
          X|-|O
          -+--+
          O|-|X
          -+--+
          X|-|O
~~~~~
Move 7:   X    5
          X|-|O
          -+--+
          O|X|X
          -+--+
          X|-|O
~~~~~
Move 8:   O    2
          X|O|O
          -+--+
          O|X|X
          -+--+
          X|-|O
~~~~~
Move 9:   X    8
          X|O|O
          -+--+
          O|X|X
          -+--+
          X|X|O
~~~~~
Winner:   Cat

```

The first line is a report label that indicates who is playing which mark and their names. The second line is blank. The following lines are in groups of 7 showing the move number, the player label, their move and the resulting tic-tac-toe board. The layout shown above must be

used, with the hyphen symbol, '-', representing unmarked squares. The board is constructed out of hyphens, pipes, '|', and plus signs. The last line must give the "Winner:" label followed by whitespace and the winning player's label and name or in the case of a tie, the string "Cat". For illegal moves a single line of output is produced as described previously. The invalid move is then ignored and processing of the following moves proceeds. It may be assumed that there are at least nine valid moves in input files with no winner and at least five valid moves in every input file.

Restrictions:

You must create an array and user-defined functions. If you do not do this, your program's score will be reduced to a zero. Also, you must produce an outline for this program and put it before main as a giant comment. It should explain in outline form what each piece does, including loops, function calls, etc.

Additionally, you must:

- Have a header comment for every major block of code describing it and its purpose.
- Use one array and at least 3 functions.
- Pass your array into at least one of these functions.
- Have a header for each function describing what it does, its preconditions, post-conditions, parameter list, specifying which parameters are input, output, and input/output.
- Not have any global variables. Doing so will greatly reduce the score of your program. Global constants are fine and in fact encouraged.

Submitting your project

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to five submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and submission link can be found at: <http://www.cs.vt.edu/curator/>

You can submit your project to the curator at: <http://eags.cs.vt.edu:8080/curator/EagsCurator>

Pledge

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
//   anyone other than my instructor or the teaching assistants  
//   assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
//   or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
//   was obtained from another source, such as a text book or course  
//   notes, that has been clearly noted with a proper citation in  
//   the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
//   interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

Failure to include this pledge in a submission is a violation of the Honor Code.