

Finish the two incomplete functions in the tic tac toe program provided. The functions are:

```
bool playerWon(char board[], char player )
```

and

```
bool catWon(char board[])
```

playerWon should return true if the character player is found in any of the 8 possible winning states.

catWon should return true when all of the 9 positions are filled with an 'X' or an 'O'.

Turn in the finished program on easel.

Click [here](#) to download the code to start with, or just copy it from below.

```
#include <iostream>
#include <math.h>
using namespace std;

const int BOARD_SIZE = 9;
void displayBoard( char board[] );
void makeMove( char board[], char turn );
bool playerWon(char board[], char player );
bool catWon(char board[]);

void main()
{
    char board[BOARD_SIZE] = {'1','2','3','4','5','6','7','8','9' };
    char turn = 'X';
    bool gameOver = false;

    do
    {
        displayBoard( board );
        makeMove( board, turn );
        //switch whose turn it is
        turn = (turn == 'X' ? 'O' : 'X');

        if(playerWon(board, 'X'))
        {
            cout << "X wins" << endl;
            gameOver = true;
        }
        if(playerWon(board, 'O'))
        {
            cout << "O wins" << endl;
            gameOver = true;
        }
        if(catWon(board))
        {
            cout << "Cat wins" << endl;
            gameOver = true;
        }
    }
}
```

```

        while( !gameOver );
    }

bool playerWon(char board[], char player )
{
    bool result = false;

    //your code goes here

    return result;
}

bool catWon(char board[])
{
    bool result = false;

    //your code goes here

    return result;
}

void makeMove( char board[], char turn )
{
    int move;

    do
    {
        cout << ">";
        cin >> move;
    }
    while( move < 1 || move > 9 || board[move-1] == 'X' ||
board[move-1] == 'O' );

    board[move-1] = turn;
}

void displayBoard( char board[] )
{
    for(int i = 0; i < BOARD_SIZE; i++)
    {
        cout << board[i];
        if( (i+1) % (int)sqrt((float)BOARD_SIZE) == 0 )
        {
            cout << endl;
        }
    }
}

```