MCS 260 Project Five : Connect-4
due Wednesday 28 November 2001, at 1PM.

The purpose of the project is to simulate the game "connect-4", also known as "4-in-line". In this game two players take turn in dropping a token in any of the slots of the vertical columns of a 7-by-7 board. After each drop the token ends up at the lowest empty square in that column on the board. The game is won by the player who first gets four tokens of his kind aligned, either horizontally, vertically, diagonally, or along the antidiagonal. The game ends in a tie when the board is filled with tokens and no player succeeded in getting four of his tokens aligned.

In this program the user plays against the computer. The strategy of the computer is very simple: it just picks a random number for the column number to make the next move.

We represent the board as a two dimensional array of characters, the token of the user is 'X', that of the computer is 'O'. Here is a screenshot :

```
Type a number between 0 and 6 to make a move : 5


 O   O X
 O   X O X
 X O X O O
 O X O O O X X
 X O X X X O X
 0 1 2 3 4 5 6
Type a number between 0 and 6 to make a move : 5
You win!
The final board :


 O   O X
 O   X O X
 X O X O O X
 O X O O O X X
 X O X X X O X
 0 1 2 3 4 5 6
```

After each move of the user, the current updated board should appear on screen. Moves are considered invalid when the column number does not lie inside the range of the board or when the column is filled up with tokens. Provide guards agains invalid moves (also make sure the computer cannot add to a full column), and allow the user to try again when the move is invalid.

The beginning of your program must contain the following preprocessor directives :

```
/* N is the size of the board, F equals "four" */
#define N 7
#define F 4
```

Use these constants consistently in the program to indicate the size of the board and the “four” so that the program keeps running fine when the constants are reset to different values, if one want to use a board of different size or change the “four” in the Connect-4.

Twenty of the forty points for the project are reserved for a correct definition of the following function :

```
int found_winner ( char b[N][N], char c );
/* returns 1 if there are F entries of c in same row, column, diagonal,
   or antidiagonal on the board b; returns 0 otherwise */
```

Correct programs that do not use any function definitions at all will be discounted with ten points. A correct program does not contain spelling or grammar mistakes in the dialogue with the user.

The first line of your program should be like

```
/* MCS 260 Project Five by <Author> */
```

where you replace the `<Author>` by your name.

The gcc compiler will be used to test your program. So even if you have developed your program in a Windows environment, test your code on `icarus.cc.uic.edu` with gcc. To ensure your program is written in ANSI C, compile with the option `-ansi`, e.g., like

```
gcc -ansi proj5.c
```

Programs that are not conform the ANSI C standard will be discounted.

The printout of your program will be collected at the start of the lecture on Wednesday 28 November, at 1PM. Before the deadline, send the code by e-mail to `janv@uic.edu` from your account on icarus. If you saved your program in the file `proj5.c`, then you send your program typing on icarus the following command :

```
mail janv@uic.edu < proj5.c
```

This ensures your program is free from any encapsulations. Note that the e-mail address is my university netid and not the address at the MSCS department. Please e-mail your program only once.

If you have questions, comments, or difficulties, feel free to come to my office for help.