#include <iostream>

using namespace std;

char matrix[3][3] = { '1', '2', '3', '4', '5', '6', '7', '8', '9' };

char player = 'X';

void Draw()

{

cout << "Tic Tac Toe By JMM" << endl;

for (int i = 0; i < 3; i++) //For x value

{

for(int j = 0; j < 3; j++) //For y value

{

cout << matrix[i][j]; //print matrix

}

cout << endl; //adds new line then prints second, now we have 3x3 square

}

}

void Input()

{

int a;

cout << "Press the number on which you want to place your token: ";

cin >> a; //take input from user, stores in a.

if (a==1)

matrix[0][0] = player;

else if (a==2)

matrix[0][1] = player;

else if (a==3)

matrix[0][2] = player;

else if (a==4)

matrix[1][0] = player;

else if (a==5)

matrix[1][1] = player;

else if (a==6)

matrix[1][2] = player;

else if (a==7)

matrix[2][0] = player;

else if (a==8)

matrix[2][1] = player;

else if (a==9)

matrix[2][2] = player; //Places token based on user input.

}

void Toggle()// determines whose turn it is.

{

if(player == 'X')

player = '0';

else

player = 'X';

}

char WinCondition()

{

if(matrix [0][0] == 'X' && matrix [0][1] == 'X' && matrix [0][2] == 'X' )//first player

return 'X';

if(matrix [1][0] == 'X' && matrix [1][1] == 'X' && matrix [1][2] == 'X' )

return 'X';

if(matrix [2][0] == 'X' && matrix [2][1] == 'X' && matrix [2][2] == 'X' )//Check for X winner in rows.

return 'X';

if(matrix [0][0] == 'X' && matrix [1][0] == 'X' && matrix [2][0] == 'X' )

return 'X';

if(matrix [0][1] == 'X' && matrix [1][1] == 'X' && matrix [2][1] == 'X' )

return 'X';

if(matrix [0][2] == 'X' && matrix [1][2] == 'X' && matrix [2][2] == 'X' )//Check for X winner in columns.

return 'X';

if(matrix [0][0] == 'X' && matrix [1][1] == 'X' && matrix [2][2] == 'X' )//Diagonal 1 winner check

return 'X';

if(matrix [2][0] == 'X' && matrix [1][1] == 'X' && matrix [0][2] == 'X' )//Diagonal 2 winner check

return 'X';

if(matrix [0][0] == 'O' && matrix [0][1] == 'O' && matrix [0][2] == 'O' )//second player

return 'O';

if(matrix [1][0] == 'O' && matrix [1][1] == 'O' && matrix [1][2] == 'O' )

return 'O';

if(matrix [2][0] == 'O' && matrix [2][1] == 'O' && matrix [2][2] == 'O' )//Check for winner in rows.

return 'O';

if(matrix [0][0] == 'O' && matrix [1][0] == 'O' && matrix [2][0] == 'O' )

return 'O';

if(matrix [0][1] == 'O' && matrix [1][1] == 'O' && matrix [2][1] == 'O' )

return 'O';

if(matrix [0][2] == 'O' && matrix [1][2] == 'O' && matrix [2][2] == 'O' )//Check for winner in columns.

return 'O';

if(matrix [0][0] == 'O' && matrix [1][1] == 'O' && matrix [2][2] == 'O' )//Diagonal 1 winner check

return 'O';

if(matrix [2][0] == 'O' && matrix [1][1] == 'O' && matrix [0][2] == 'O' )//Diagonal 2 winner check

return 'O';

return '/';

// 00 01 02

// 10 11 12

// 20 21 22 Board reference

}

int main()

{

Draw();

while(1)//continuous, never ends. Always checks turn.

{

Input();

Draw();

if (WinCondition()== 'X')//Win conditions met

{

cout << "X Wins!";

break;

}

else if (WinCondition() == 'O')

{

cout << "O Wins!";

break;

}

Toggle();

}

system("pause");

return 0;

}