// **Simple tic-tac-toe with minimax**

```
#include <iostream>
using namespace std;

void makemove();
int min(int depth);
int max(int depth);
int evaluate();
int check4winner();
void checkGameOver();
void getamove();
void setup();
void printboard();

int b[3][3], maxdepth=9;

int main ()
{ setup();
  printboard();
  for (;;)
  {  getamove();
     checkGameOver();
     makemove();
     checkGameOver();
} }

void printboard()
{ cout << endl;
  cout << b[0][0] << " " << b[0][1] << " " << b[0][2] << endl;
  cout << b[1][0] << " " << b[1][1] << " " << b[1][2] << endl;
  cout << b[2][0] << " " << b[2][1] << " " << b[2][2] << endl;
}

void setup()
{ for (int i=0; i<3; i++) for (int j=0; j<3; j++) { b[i][j]=0; } }

void getamove()
{ int i,j;
  cout << "Enter your move:  ";
  cin >> i >> j;
  b[i][j]=2;
}

int evaluate ()
{ return 0; }


void makemove()
{ int best=-20000,depth=maxdepth,score,mi,mj;
  for (int i=0; i<3; i++)
  { for (int j=0; j<3; j++)
    { if (b[i][j]==0)
      { b[i][j]=1; // make move on board
        score = min(depth-1);
        if (score > best) { mi=i; mj=j; best=score; }
        b[i][j]=0; // undo move
  } } }
  cout << "my move is " << mi << " " << mj << endl;
  b[mi][mj]=1;
}

int min(int depth)
{ int best=20000,score;
  if (check4winner() != 0) return (check4winner());
  if (depth == 0) return (evaluate());
  for (int i=0; i<3; i++)
  { for (int j=0; j<3; j++)
    { if (b[i][j]==0)
      { b[i][j]=2; // make move on board
        score = max(depth-1);
        if (score < best) best=score;
        b[i][j]=0; // undo move
  } } }
  return(best);
}

int max(int depth)
{ int best=-20000,score;
  if (check4winner() != 0) return (check4winner());
  if (depth == 0) return (evaluate());
  for (int i=0; i<3; i++)
  { for (int j=0; j<3; j++)
    { if (b[i][j]==0)
      { b[i][j]=1; // make move on board
        score = min(depth-1);
        if (score > best) best=score;
        b[i][j]=0; // undo move
  } } }
  return(best);
}

int check4winner()
{ if ((b[0][0]==1)&&(b[0][1]==1)&&(b[0][2]==1)
   || (b[1][0]==1)&&(b[1][1]==1)&&(b[1][2]==1)
   || (b[2][0]==1)&&(b[2][1]==1)&&(b[2][2]==1)
   || (b[0][0]==1)&&(b[1][0]==1)&&(b[2][0]==1)
   || (b[0][1]==1)&&(b[1][1]==1)&&(b[2][1]==1)
   || (b[0][2]==1)&&(b[1][2]==1)&&(b[2][2]==1)
   || (b[0][0]==1)&&(b[1][1]==1)&&(b[2][2]==1)
   || (b[0][2]==1)&&(b[1][1]==1)&&(b[2][0]==1)) return 5000;  //
computer wins
  if ((b[0][0]==2)&&(b[0][1]==2)&&(b[0][2]==2)
   || (b[1][0]==2)&&(b[1][1]==2)&&(b[1][2]==2)
   || (b[2][0]==2)&&(b[2][1]==2)&&(b[2][2]==2)
   || (b[0][0]==2)&&(b[1][0]==2)&&(b[2][0]==2)
   || (b[0][1]==2)&&(b[1][1]==2)&&(b[2][1]==2)
   || (b[0][2]==2)&&(b[1][2]==2)&&(b[2][2]==2)
   || (b[0][0]==2)&&(b[1][1]==2)&&(b[2][2]==2)
   || (b[0][2]==2)&&(b[1][1]==2)&&(b[2][0]==2)) return -5000;
  for (int i=0; i<3; i++)
    for (int j=0; j<3; j++)
      {if (b[i][j]==0) return 0;}
  return 1; // draw
}

void checkGameOver()
{ printboard();
  if (check4winner() == -5000) { cout << "you win" << endl; exit(0); }
  if (check4winner() == 5000)  { cout << "I win"   << endl; exit(0); }
  if (check4winner() == 1)     { cout << "draw"    << endl; exit(0); }
}
```