Jumping Balls

Have you ever played those old fashioned wood puzzles with divots cut into a piece of wood for marbles to sit in? This is one of those games. Try to get the Xs on one side to swap places with the Os on the other. Pieces can move a space or jump over a piece. Since there's always one and only one empty space all you need to do is decide which marble to move.

Not only is this short program a fun solitaire game to play but the way the game plays on the screen, stacking the boards on top of each other, the optimum solution (24 moves) makes a pretty pattern that looks maybe like a friendship bracelet. When you beat it you'll see it.

Jumping balls was written by Joseph Larson, inspired by a BASIC game of the same name by Anthony Rizzolo as found in 'More BASIC Computer Games' edited by David H. Ahl (c) 1979.

NAME.C	You will need: a C/C++ complier .

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <ctype.h>

int main (void) {
  char yn, b[10], *start = "XXXX.OOOO", *end = "OOOO.XXXX";
  int empty, input, t;

  puts ("Jumping Balls\n------- -----\n"
  "This is a game of solitare. It is played with 8 balls in 9 holes. The 
X's\n"
  "and O's are the balls in this game and the period ('.') is the empty 
hole.\n"
  "The object is to get all the X's and O's to switch sides. You can move 
by\n"
  "either moving a ball into the empty space or jumping one ball over an-
other\n"
  "into the empty space.\n\n"
  "On your turn input the number from the left of the space of the ball 
you\n"
  "want to move or jump into the empty space.\n\n"
  "If you want to quit input '0' as your move.\n\n"
  "Press ENTER to start...");
  getchar ();
  do {
    empty = 4; t = 0; strcpy (b, start);
    puts ("\t123456789");
    printf ("\t%s\t? ", b); scanf ("%d", &input);
    while (input && strcmp (b, end)) {
      if (input--) {
        if (input >= 0 && input < 9 &&
          (abs (input - empty) == 2 || abs (input - empty) == 1)) {
          b[empty] = b[input]; b[input] = '.'; empty = input; t ++;
        } else puts ("Invalad move.");
      }
      printf ("\t%s\t? ", b); if (strcmp (b, end)) scanf ("%d", &input);
    }
    if (!strcmp (b, end))
      printf ("\nCongratulations! You solved it in %d moves!\n\n"
      "Do you want to try for a better score? (y/n) ", t);
    else printf ("Do you want to start again ? (y/n) ");
    do yn = getchar (); while (!isalpha (yn));
  } while (tolower (yn) != 'n');
  puts ("Thanks for playing!");
  exit (0);
}
```