```
// Samson Fung

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

int w, x, y, z;
int gamescount;
int lengthparameter, n;
int BOARD[9];
int WinXTotal, WinOTotal, TieGameTotal, UnfinishedTotal;

void ResetBoardandVariables();
void loadBOARDFromStream();
void displayGame(int arg[], int lengthparameter);
int WinX();
int WinY();
int getGameStatus();
int displayTotals();

void loadBOARDsFromStream()
    {
        std::ifstream file;
        file.open("games.txt");
        while (!file.eof())
            {
            cout << endl;
            for (z=0; z<9; ++z)
            {
                file >> x >> y;
                w = (x + 3*(y-1));
                if (x != 0 or y != 0)
                    {if(z%2 == 0)
                        BOARD[w-1] = 1;
                    else
                        BOARD[w-1] = 2;}
                else
                {
                    z = 10, gamescount++;
                    getGameStatus();
                    displayGame(BOARD, 9);
                    int BOARD[9] = {0,0,0,0,0,0,0,0,0};
                }
            }
            }
        file.close();
    }

void displayGame(int arg[], int lengthparameter)
    {
        for (n=0; n<lengthparameter; ++n)
            {
            cout << "  ";
            if (arg[n] == 0)
                cout << " ";
            if (arg[n] == 1)
                cout << 'X';
            if (arg[n] == 2)
                cout << 'O';
            cout << "  ";
            if ((n+1)%3 == 0 && n>1)
                cout << endl << "- - - - - - - - -" << endl;
            else cout << '|';
            }
    }

int WinX()
{
    if (BOARD[0] == BOARD[1] && BOARD[1] == BOARD[2] && BOARD[2] == 1)
        return 1;
    else if (BOARD[3] == BOARD[4] && (BOARD[4] == BOARD[5] == 1))
        return 1;
    else if (BOARD[6] == BOARD[7] && BOARD[7] == BOARD[8] && BOARD[8] == 1)
        return 1;
    else if (BOARD[0] == BOARD[3] && BOARD[3] == BOARD[6] && BOARD[6] == 1)
        return 1;
    else if (BOARD[1] == BOARD[4] && BOARD[4] == BOARD[7] && BOARD[7] == 1)
```

```
 78          return 1;
 79      else if (BOARD[2] == BOARD[5] && BOARD[5] == BOARD[8] && BOARD[8] == 1)
 80          return 1;
 81      else if (BOARD[0] == BOARD[4] && BOARD[4] == BOARD[8] && BOARD[8] == 1)
 82          return 1;
 83      else if (BOARD[2] == BOARD[4] && BOARD[4] == BOARD[6] && BOARD[6] == 1)
 84          return 1;
 85      else return 0;
 86  }
 87
 88  int WinO()
 89  {
 90      if (BOARD[0] == BOARD[1] && BOARD[1] == BOARD[2] && BOARD[2] == 2)
 91          return 2;
 92      else if (BOARD[3] == BOARD[4] && BOARD[4] == BOARD[5] && BOARD[5] == 2)
 93          return 2;
 94      else if (BOARD[6] == BOARD[7] && BOARD[7] == BOARD[8] && BOARD[8] == 2)
 95          return 2;
 96      else if (BOARD[0] == BOARD[3] && BOARD[3] == BOARD[6] && BOARD[6] == 2)
 97          return 2;
 98      else if (BOARD[1] == BOARD[4] && BOARD[4] == BOARD[7] && BOARD[7] == 2)
 99          return 2;
100      else if (BOARD[2] == BOARD[5] && BOARD[5] == BOARD[8] && BOARD[8] == 2)
101          return 2;
102      else if (BOARD[0] == BOARD[4] && BOARD[4] == BOARD[8] && BOARD[8] == 2)
103          return 2;
104      else if (BOARD[2] == BOARD[4] && BOARD[4] == BOARD[6] && BOARD[6] == 2)
105          return 2;
106      else return 0;
107  }
108
109  int Tie()
110      {
111          if (BOARD[1] != 0 && BOARD[2] != 0 && BOARD[3] != 0
112                      && BOARD[4] != 0 && BOARD[5] != 0 && BOARD[6] != 0
113                    && BOARD[7] != 0 && BOARD[8] != 0 && BOARD[9] != 0)
114          return 3;
115          else return 0;
116      }
117
118  int getGameStatus()
119      {
120          if (WinX() == 1 or WinO() == 2 or Tie() == 3)
121              {
122              if (WinX() == 1)
123                  {
124                      ++WinXTotal;
125                      cout << "Game " << gamescount << " is a win for X." << endl;
126                  }
127              else
128              if (WinO() == 2)
129                  {
130                      ++WinOTotal;
131                      cout << "Game " << gamescount << " is a win for O." << endl;
132                  }
133              else
134              if (Tie() == 3)
135                  {
136                      ++TieGameTotal;
137                      cout << "Game " << gamescount << " is a tie." << endl;
138                  }
139              }
140          else
141              {
142                  ++UnfinishedTotal;
143                  cout << "Game " << gamescount << " is unfinished." << endl;
144              }
145      }
146
147  int displayTotals()
148      {
149          cout << endl;
150          cout << "X has won " << WinXTotal << " times." << endl;
151          cout << "O has won " << WinOTotal << " times." << endl;
152          cout << "Total tie games: " << TieGameTotal << "." << endl;
153          cout << "Total unfinished games: " << UnfinishedTotal << "." << endl;
154          cout << "Total games: " << WinXTotal+WinOTotal+TieGameTotal+UnfinishedTotal <<
```

```
    "." << endl;
155     }
156
157 int main()
158 {
159     loadBOARDsFromStream();
160     displayTotals();
161     return 0;
162 }
163
```