**#include <iostream>**

**using namespace std;**

**// Global Initializations...**

**static char grid[3][3];**

**static char one = grid[0][0] = '1';**

**static char two = grid[0][1] = '2';**

**static char three = grid[0][2] = '3';**

**static char four = grid[1][0] = '4';**

**static char five = grid[1][1] = '5';**

**static char six = grid[1][2] = '6';**

**static char seven = grid[2][0] = '7';**

**static char eight = grid[2][1] = '8';**

**static char nine = grid[2][2] = '9';**

**bool doneOne = false, doneTwo = false, doneThree = false, doneFour = false, doneFive = false, doneSix = false, doneSeven = false, doneEight = false;**

**bool doneNine = false;**

**// Enters Data into The Particular Cell..**

**void takeMove(int pm, int n)**

**{**

**switch (pm)**

**{**

**case 1:**

**if (!doneOne)**

**{**

**if (n == 1)**

**grid[0][0] = 'X';**

**if (n == 2)**

**grid[0][0] = 'O';**

**doneOne = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 2:**

**if (!doneTwo)**

**{**

**if (n == 1)**

**grid[0][1] = 'X';**

**else if (n == 2)**

**grid[0][1] = 'O';**

**doneTwo = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 3:**

**if (!doneThree)**

**{**

**if (n == 1)**

**grid[0][2] = 'X';**

**else if (n == 2)**

**grid[0][2] = 'O';**

**doneThree = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 4:**

**if (!doneFour)**

**{**

**if (n == 1)**

**grid[1][0] = 'X';**

**else if (n == 2)**

**grid[1][0] = 'O';**

**doneFour = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 5:**

**if (!doneFive)**

**{**

**if (n == 1)**

**grid[1][1] = 'X';**

**else if (n == 2)**

**grid[1][1] = 'O';**

**doneFive = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 6:**

**if (!doneSix)**

**{**

**if (n == 1)**

**grid[1][2] = 'X';**

**else if (n == 2)**

**grid[1][2] = 'O';**

**doneSix = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 7:**

**if (!doneSeven)**

**{**

**if (n == 1)**

**grid[2][0] = 'X';**

**else if (n == 2)**

**grid[2][0] = 'O';**

**doneSeven = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 8:**

**if (!doneEight)**

**{**

**if (n == 1)**

**grid[2][1] = 'X';**

**else if (n == 2)**

**grid[2][1] = 'O';**

**doneEight = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**case 9:**

**if (!doneNine)**

**{**

**if (n == 1)**

**grid[2][2] = 'X';**

**else if (n == 2)**

**grid[2][2] = 'O';**

**doneNine = true;**

**}**

**else**

**{**

**cout << '\n';**

**cout << "Cell Already Selected...Please Choose Any Other Cell.." << '\n';**

**}**

**break;**

**default:**

**cout << '\n';**

**cout << "Please Select Between 1 - 9 Cells..." << '\n';**

**}**

**}**

**// Checks if any of the Columns or Rows are Equal...**

**bool checkStatus()**

**{**

**if (grid[0][0] == grid[0][1] && grid[0][1] == grid[0][2])**

**return 1;**

**if (grid[1][0] == grid[1][1] && grid[1][1] == grid[1][2])**

**return 1;**

**if (grid[2][0] == grid[2][1] && grid[2][1] == grid[2][2])**

**return 1;**

**if (grid[0][0] == grid[1][0] && grid[1][0] == grid[2][0])**

**return 1;**

**if (grid[0][1] == grid[1][1] && grid[1][1] == grid[2][1])**

**return 1;**

**if (grid[0][2] == grid[1][2] && grid[1][2] == grid[2][2])**

**return 1;**

**if (grid[0][0] == grid[1][1] && grid[1][1] == grid[2][2])**

**return 1;**

**if (grid[0][2] == grid[1][1] && grid[1][1] == grid[2][0])**

**return 1;**

**else**

**return 0;**

**}**

**// Resets All the Grids to its Original Position....Used to Restart the Game...**

**void resetGrids()**

**{**

**grid[0][0] = '1';**

**grid[0][1] = '2';**

**grid[0][2] = '3';**

**grid[1][0] = '4';**

**grid[1][1] = '5';**

**grid[1][2] = '6';**

**grid[2][0] = '7';**

**grid[2][1] = '8';**

**grid[2][2] = '9';**

**doneOne = false;**

**doneTwo = false;**

**doneThree = false;**

**doneFour = false;**

**doneFive = false;**

**doneSix = false;**

**doneSeven = false;**

**doneEight = false;**

**doneNine = false;**

**}**

**// Checks if all cells are not equal...used when a match is // tied...**

**bool stat()**

**{**

**if (grid[0][0] != '1' && grid[0][1] != '2' && grid[0][2] != '3' && grid[1][0] != '4' && grid[1][1] != '5' && grid[1][2] != '6' && grid[2][0] != '7' && grid[2][1] != '8' && grid[2][2] != '9')**

**return 1;**

**else**

**return 0;**

**}**

**int main()**

**{**

**char ch;**

**resetGrids();**

**int pm, n;**

**static int count, flag;**

**count = 0;**

**flag = 0;**

**while (1)**

**{**

**cout << '\n';**

**cout << "WARNING : Wrong Selection Also Costs You a CHANCE..." << '\n';**

**cout << '|' << " " << grid[0][0] << " " << '|' << " " << grid[0][1] << " " << '|' << " " << grid[0][2] << " " << '|' << " " << '\n';**

**cout << "--------------" << '\n';**

**cout << '|' << " " << grid[1][0] << " " << '|' << " " << grid[1][1] << " " << '|' << " " << grid[1][2] << " " << '|' << " " << '\n';**

**cout << "--------------" << '\n';**

**cout << '|' << " " << grid[2][0] << " " << '|' << " " << grid[2][1] << " " << '|' << " " << grid[2][2] << " " << '|' << " " << '\n';**

**cout << '\n';**

**cout << "Enter Player 1's Move : (1-9) " << '\n';**

**cin >> pm;**

**n = 1;**

**takeMove(pm, n);**

**if (stat())**

**{**

**cout << "Match Tied!!";**

**break;**

**}**

**if (checkStatus())**

**{**

**cout << '\n';**

**cout << "Player ONE Wins!!" << '\n';**

**cout << '\n';**

**flag = 1;**

**break;**

**}**

**cout << '\n';**

**cout << '|' << " " << grid[0][0] << " " << '|' << " " << grid[0][1] << " " << '|' << " " << grid[0][2] << " " << '|' << " " << '\n';**

**cout << "--------------" << '\n';**

**cout << '|' << " " << grid[1][0] << " " << '|' << " " << grid[1][1] << " " << '|' << " " << grid[1][2] << " " << '|' << " " << '\n';**

**cout << "--------------" << '\n';**

**cout << '|' << " " << grid[2][0] << " " << '|' << " " << grid[2][1] << " " << '|' << " " << grid[2][2] << " " << '|' << " " << '\n';**

**cout << '\n';**

**cout << "Enter Player 2's Move : (1-9) " << '\n';**

**cin >> pm;**

**n = 2;**

**takeMove(pm, n);**

**if (stat())**

**{**

**cout << "Match Tied!!" << '\n';**

**break;**

**}**

**if (checkStatus())**

**{**

**cout << '\n';**

**cout << "Player TWO Wins!!" << '\n';**

**flag = 1;**

**cout << '\n';**

**break;**

**}**

**}**

**cout << "Restart Game ?? (Y/N)" << '\n';**

**cin >> ch;**

**if (ch == 'Y' || ch == 'y')**

**main();**

**else**

**{**

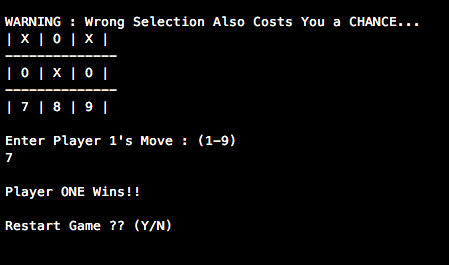
**cout << "Thank You..." << '\n';**

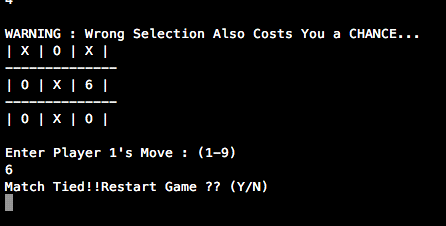
**exit(0);**

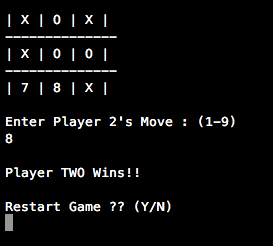
**}**

**}**

**Output :**

****

****

****