//Shrabanti Basu

//Program 9

//April 14, 2016

//This program develops a interactive tic-tac toe game using a two dimensional array, loops and conditions,

//functions and function calls

#include <iostream>

#include <cstdlib>

using namespace std;

//function prototypes

void drawboard(char arr[3][3]);

void getInput(char arr[3][3], char);

char isWin(char arr[3][3]);

const int ROWS = 3, COLS = 3; //declare array dimensions as constant variables

int main()

{

//print information

cout << "Shrabanti Basu\n";

cout << "Program 9\n";

cout << "April 14, 2016\n";

cout << "This program generates an interactive tic-tac toe game using a 2-D array,\n"

<< "loops and conditions, and functions.\n\n";

cout << "The Basics: \n\n";

cout << "The program displays a board for tic-tac-toe game, with numbers indicating positions.\n"

<< "Choose a valid symbol and then enter the position where you want to place your symbol.\n"

<< "The program then toggles symbol for the other player to take turn and so on.\n\n"

<< "The first player will choose either X or O, and the second player will be assigned symbol\n"

<< "accordingly. The current game continues till one of the players win, or all the\n"

<< "nine places get filled, and a draw results.\n";

cout << "Also, the game will not let the user select a position that is already\n"

<< "occupied with an X or an O.\n\n";

//Pause the program for the user to read information

//When user hits the Enter key, the drawBoard() is called

//We stop the program at this point so the system does not clear

//and the user is able to read the information

cout << "Please press enter to continue to the tic tac toe game.";

cin.get();

int counter; //counter for loop iteration

char winner; //to store symbol of the player who wins returned from isWin(), or a draw

char choice; //choice of user to repeat or not

char symbol = 'X'; //user's choice of symbol in turn taking and initialized to'X'

char board[ROWS][COLS]; //declare the 2D array

do

{

counter = 0; //set counter to zero

//initialize the array with values that will show the places to the user

board[0][0] = '1';

board[0][1] = '2';

board[0][2] = '3';

board[1][0] = '4';

board[1][1] = '5';

board[1][2] = '6';

board[2][0] = '7';

board[2][1] = '8';

board[2][2] = '9';

//draw the game board

drawboard(board);

//ask user what symbol he/she wants to choose

cout << "Please select your symbol (X or O (the letter 'O', not zero;\n"

<< "and all uppercase) only): ";

cin >> symbol;

//repeat asking for a valid symbol

while (!(symbol == 'X' || symbol == 'O'))

{

cout << "That was not a valid symbol.\n"

<< "Please select your symbol (X or O only): ";

cin >> symbol;

}

while (counter < 9)

{

//get user input

getInput(board, symbol);

//draw board again

drawboard(board);

//change player by alternating symbol

if (symbol == 'X')

symbol = 'O';

else

symbol = 'X';

//check for a win situation by calling isWin() function

//if a player wins, break the current iteration and start over a new game

winner = isWin(board);

if (winner == 'X')

{

cout << "Player X wins.\n\n";

break;

}

else if (winner == 'O')

{

cout << "Player O wins.\n\n";

break;

}

counter++; //increment counter

}

//if there is no winner, that is if the value in winner is

//not equal to 'X' or 'O', then it will be a tie.

if (winner == 'T')

cout << "Nobody wins. It is a tie...\n\n";

//ask user if he/she wants to repeat

cout << "Want to play again? Enter Y for yes, N for no: ";

cin >> choice;

} while (choice == 'Y' || choice == 'y');

cout << "Thanks for playing the game.\n";

return 0;

}

//function definition to draw the game board with proper lines

//the drawboard function takes a 2D array as its parameter and prints it

//also it clears the previous screen everytime

void drawboard(char arr[3][3])

{

system("cls"); //clear previous screen

//print the board to console

for (int i = 0; i < ROWS; i++)

{

for (int j = 0; j < COLS; j++)

{

cout << " " << arr[i][j] << " | ";

}

cout << "\n -------------" << endl;

}

}

//Definition of getInput function

//it takes a 2D array and a char variable as parameters

//the function asks user for a valid number

//the user input determines where the character passed through sym

//will be put in the array

//if a valid number is not entered, or the place is already filled

//the getInput() function calls itself again

void getInput(char arr[3][3], char sym)

{

int input; //to store position in the array

cout << "Enter the field number for " << sym << " : ";

cin >> input;

//check for valid positions and put symbols there accordingly

//number 1 is position [0][0] in array

//number 2 is position [0][1] in array, etc

//after each three elements we go to the next row in the array

switch (input)

{

case 1:

if (arr[0][0] == 'X' || arr[0][0] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[0][0] = sym;

break;

case 2:

if (arr[0][1] == 'X' || arr[0][1] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[0][1] = sym;

break;

case 3:

if (arr[0][2] == 'X' || arr[0][2] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[0][2] = sym;

break;

case 4:

if (arr[1][0] == 'X' || arr[1][0] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[1][0] = sym;

break;

case 5:

if (arr[1][1] == 'X' || arr[1][1] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[1][1] = sym;

break;

case 6:

if (arr[1][2] == 'X' || arr[1][2] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[1][2] = sym;

break;

case 7:

if (arr[2][0] == 'X' || arr[2][0] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[2][0] = sym;

break;

case 8:

if (arr[2][1] == 'X' || arr[2][1] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[2][1] = sym;

break;

case 9:

if (arr[2][2] == 'X' || arr[2][2] == 'O')

{

cout << "Place already full. Enter a different position.\n";

getInput(arr, sym);

}

else

arr[2][2] = sym;

break;

default:

cout << "Not a valid number.\n";

getInput(arr, sym);

break;

}

}

//Definition of getInput function

//it takes a 2D array as its parameter

//and checks for a win or draw situation.

//when any row or column or diagonal has all the same symbols,

//that player wins, if not it is a draw

//in case of a win, the function returns either 'X', or 'O',

//depending on the players, else returns 'D' which is a draw

//the if-else conditions check for every row, column or diagonal in the array for a winning situation

char isWin(char arr[3][3])

{

char win = 'T'; //to store sysmbol to winner, and initialize it to 'D'

//see if X wins by checking all rows, all columns and all diagonals

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

win = 'X';

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

win = 'X';

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

win = 'X';

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

win = 'X';

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

win = 'X';

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

win = 'X';

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

win = 'X';

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

win = 'X';

//see if O wins by checking all rows, all columns and all diagonals

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

win = 'O';

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

win = 'O';

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

win = 'O';

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

win = 'O';

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

win = 'O';

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

win = 'O';

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

win = 'O';

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

win = 'O';

return win;

}