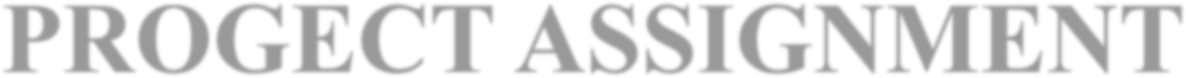


**Programming**

**fundamentals**

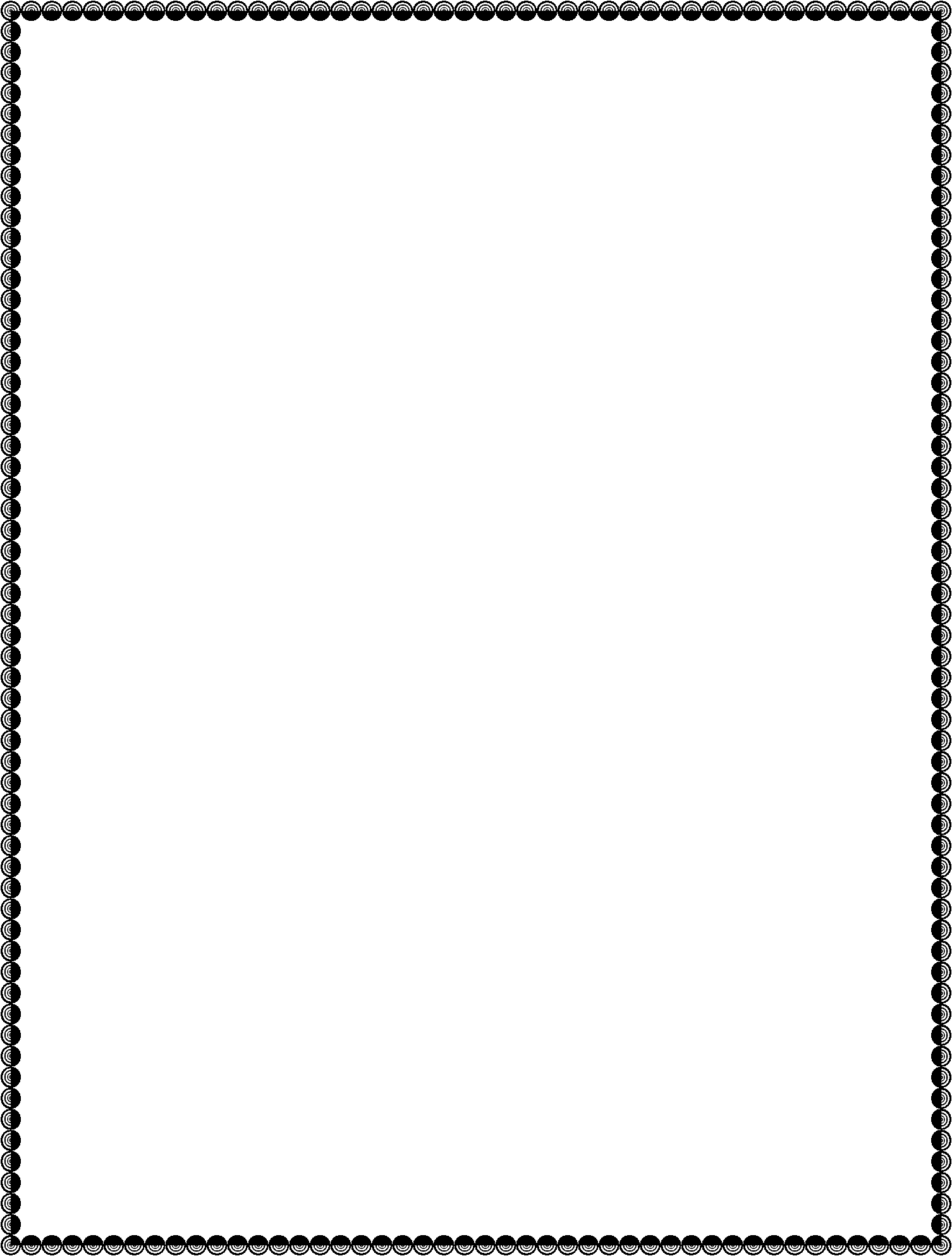


**PROGECT ASSIGNMENT**

# TIC TAC TOE GAME

# DEMO

#include<iostream> #include <cstdlib> #include<string> #include<ctime> #include<fstream> using namespace std;



struct Player { char symbol; string name; int wins;

int user\_mode;

};

char board[3][3] = {{'1','2','3'}, {'4','5','6'}, {'7','8','9'}};

char turn = 'X', user\_ask;

int row, column,total\_game, player\_x\_winner = 0, player\_o\_winner = 0, tie\_game

= 0;

bool draw = false;

Player player\_x, player\_o;

//for design ============ void printdesign(){

for(int i=0;i<=45;i++){ cout<<"=";

}

}

//for design ------------ void printminus(){

for(int i=0;i<=45;i++){ cout<<"-";

}

}

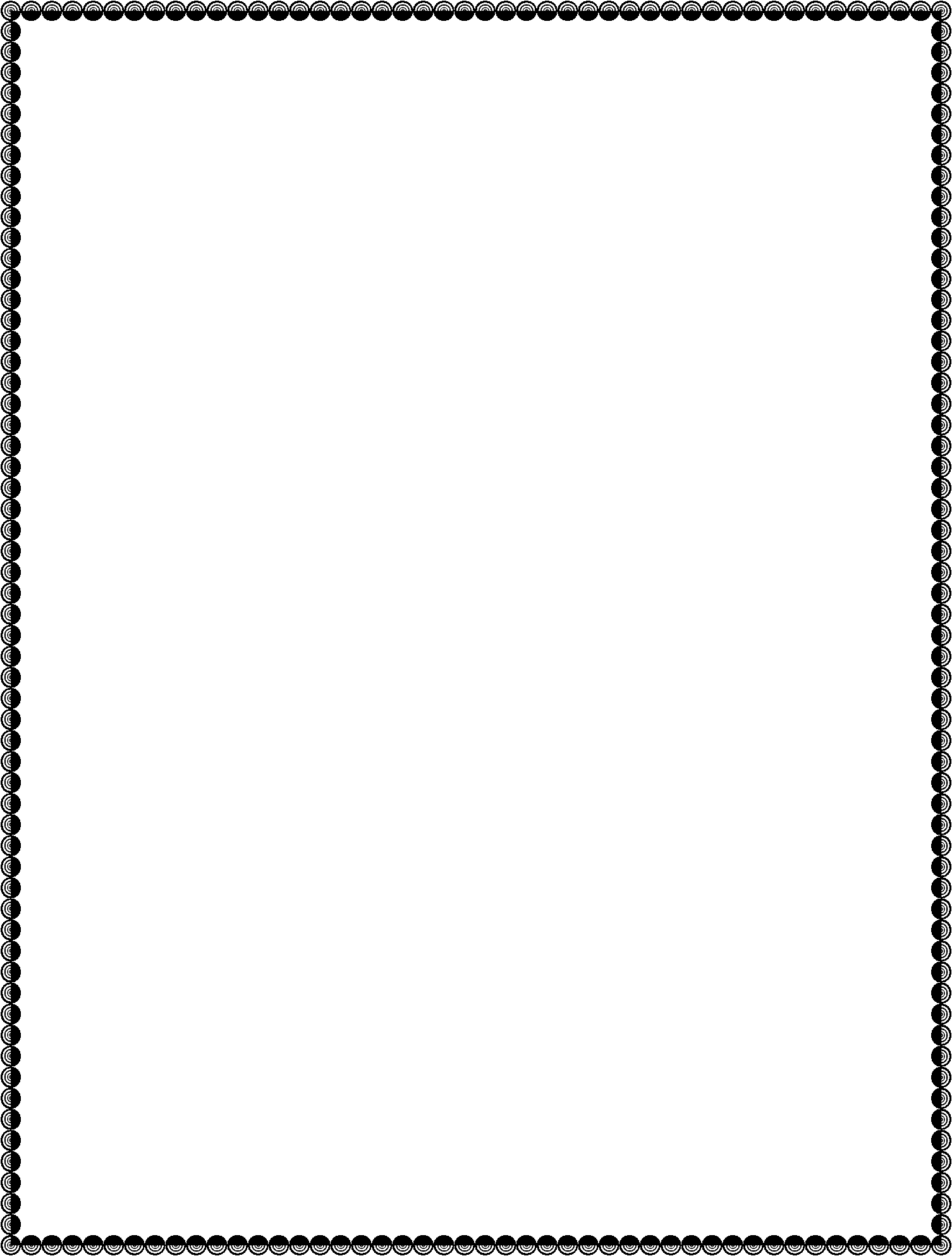
void game\_label(){ printdesign(); cout<<endl;

cout << "\n\t\tTIC TAC TOE GAME" << endl; cout<<endl;

printdesign();

}

void display\_board(){ system("cls"); if(player\_x.user\_mode==1){



game\_label(); cout<<endl;

cout << "\n\t PLAYER1 [X] Vs PLAYER2 [O]\n\n";

}else if(player\_x.user\_mode==2){ game\_label();

cout<<endl;

cout << "\n\t Computer [O] Vs Player [X]\n\n";

}

cout<<endl;

cout << "\t\t | | \n";

cout << "\t\t " << board[0][0] << " | " << board[0][1] << " | " << board[0][2] << "\n";

cout<<"\t\t | | \n"; cout << "\t\t | | \n";

cout << "\t\t " << board[1][0] << " | " << board[1][1] << " | " << board[1][2] << "\n";

cout<<"\t\t | | \n"; cout << "\t\t | | \n";

cout << "\t\t " << board[2][0] << " | " << board[2][1] << " | " << board[2][2] << "\n";

cout << "\t\t | | \n";

}

void player\_turn() { int choice;

if (turn == 'X') {

cout << "\n\tPLAYER1 [X] named \"" << player\_x.name << "\" turn : ";

} else if (turn == 'O') {

cout << "\n\tPLAYER2 [O] named \"" << player\_o.name << "\" turn : ";

}

cin >> choice;

// Validate input range

if (choice < 1 || choice > 9) {

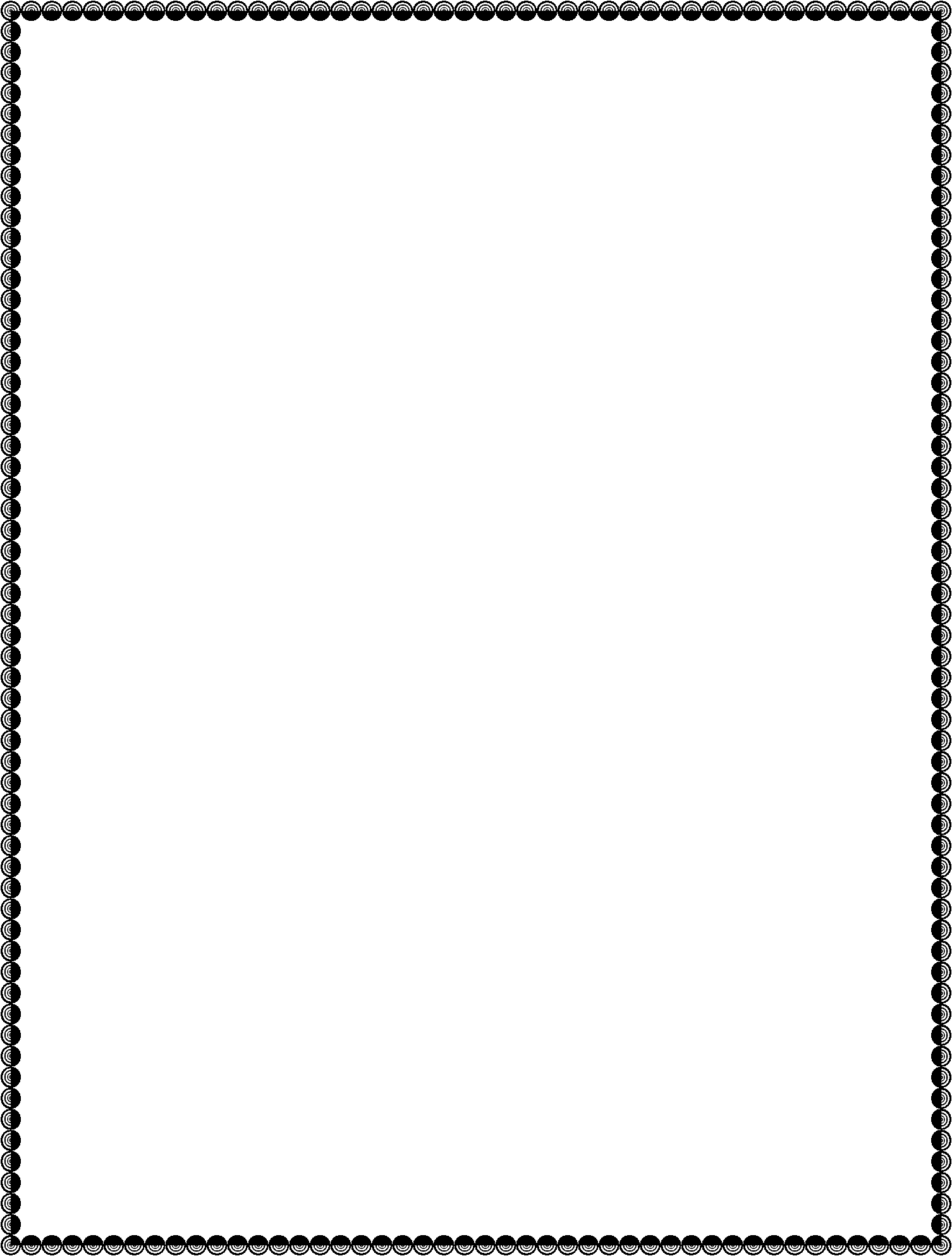
cout << "\tInvalid choice. Please enter a number between 1 and 9.\n"; player\_turn();

return;

}

// Convert choice to row and column row = (choice - 1) / 3;

column = (choice - 1) % 3;



// Check if the chosen cell is already taken

if (board[row][column] == 'X' || board[row][column] == 'O') {

cout << " Box already filled\n Please enter another number\n";

player\_turn(); return;

}

// Place the symbol and switch turn if (turn == 'X') {

board[row][column] = 'X'; turn = 'O';

} else {

board[row][column] = 'O'; turn = 'X';

}

display\_board();

}

void computer\_turn() {

// Basic computer move logic (random move) srand(time(0));

int choice; do {

choice = rand() % 9 + 1; // Generate a random number between 1 and 9 switch (choice) {

case 1: row = 0; column = 0; break;

case 2: row = 0; column = 1; break;

case 3: row = 0; column = 2; break;

case 4: row = 1; column = 0; break;

case 5: row = 1; column = 1; break;

case 6: row = 1; column = 2; break;

case 7: row = 2; column = 0; break;

case 8: row = 2; column = 1; break;

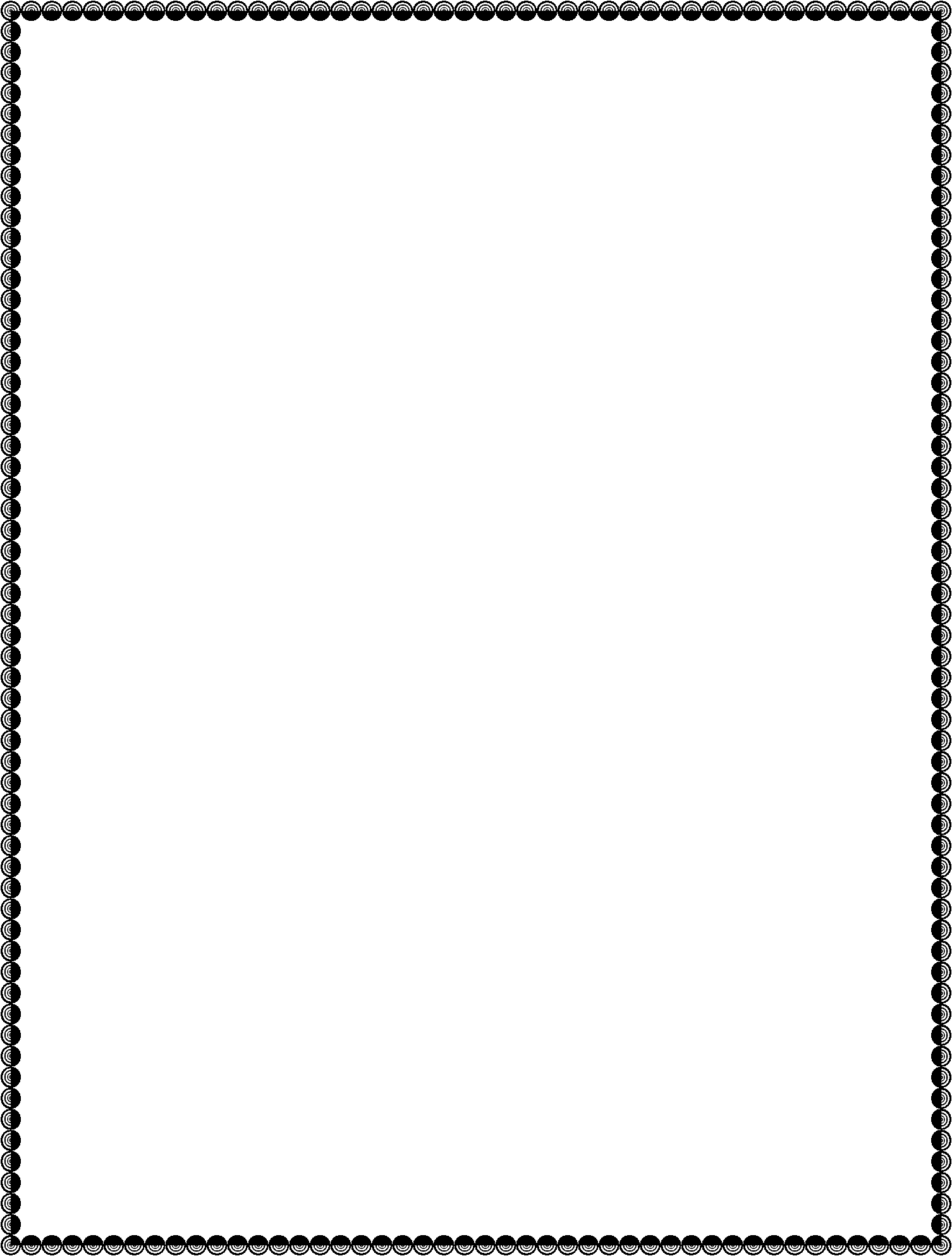
case 9: row = 2; column = 2; break;

}

} while (board[row][column] == 'X' || board[row][column] == 'O'); // Check if the chosen BLOCK is already taken

cout << "\n\nComputer [O] turns: " << choice << endl; board[row][column] = 'O'; // The computer's move turn = 'X'; // Switch turn to the player [x]

}



bool game\_over(){

// Check which player wins the game for(int i=0; i<3; ++i){

// Check current row

if (board[i][0] == board[i][1] && board[i][0] == board[i][2]) return false;

// Check current column

if(board[0][i] == board[1][i] && board[0][i] == board[2][i]) return false;

}

// Check left top to bottom diagonal

if(board[0][0] == board[1][1] && board[0][0] == board[2][2]) return false;

// Check right top to bottom diagonal

if(board[0][2] == board[1][1] && board[0][2] == board[2][0]) return false;

// Check if all boxes are not filled for(int i=0; i<3; ++i){

for(int j=0; j<3; ++j){

if(board[i][j] != 'X' && board[i][j] != 'O') return true;

}

}

// game draw

draw = true; return false;

}

void scoreboard(){

//create file

ofstream file("tictactoe.txt"); if(!file)

{

cout<<"File not created.\n";

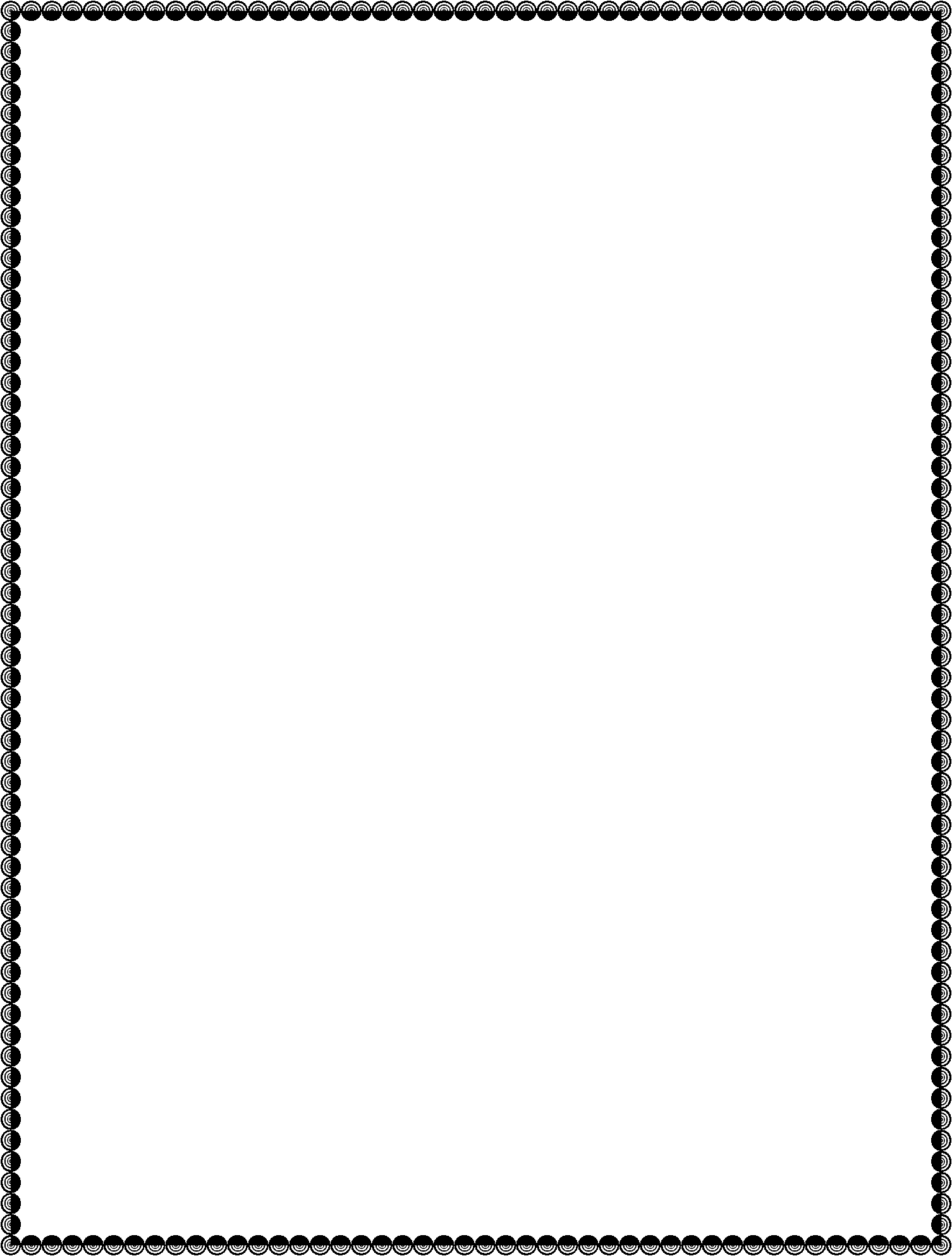
}

else

{

cout<<"File created successfuly.\n";

system("cls"); printdesign(); cout<<"\n\t\tScoreboard\n"; printdesign();



cout << endl; if(player\_x.user\_mode==2){

cout<<"\t Computer[O] VS Player[X] ";

file<<" ";

file<<" Computer[O] VS Player[X] ";

file<<" ";

cout<<endl; printdesign(); cout<<endl;} else{

cout<<"\t Player[X] VS Player[O] ";

file<<" ";

file<<" Player[X] VS Player[O] ";

file<<" ";

cout<<endl; printdesign(); cout<<endl;

}

cout << endl; total\_game=player\_x.wins+player\_o.wins+tie\_game; cout<<endl<<"Total games played: "<<total\_game; cout<<endl;

file<<endl<<"Total games played: "<<total\_game;

cout << "\nPLAYER1 [X] named \"" << player\_x.name << "\" wins: " << player\_x.wins;

cout << endl;

file<<"\nPLAYER1 [X] named \"" << player\_x.name << "\" wins: " << player\_x.wins;

if(player\_x.user\_mode==2){

cout<<endl<< player\_o.name<<" [O] wins: "<<player\_o.wins; file<<endl<< player\_o.name<<" [O] wins: "<<player\_o.wins;

}

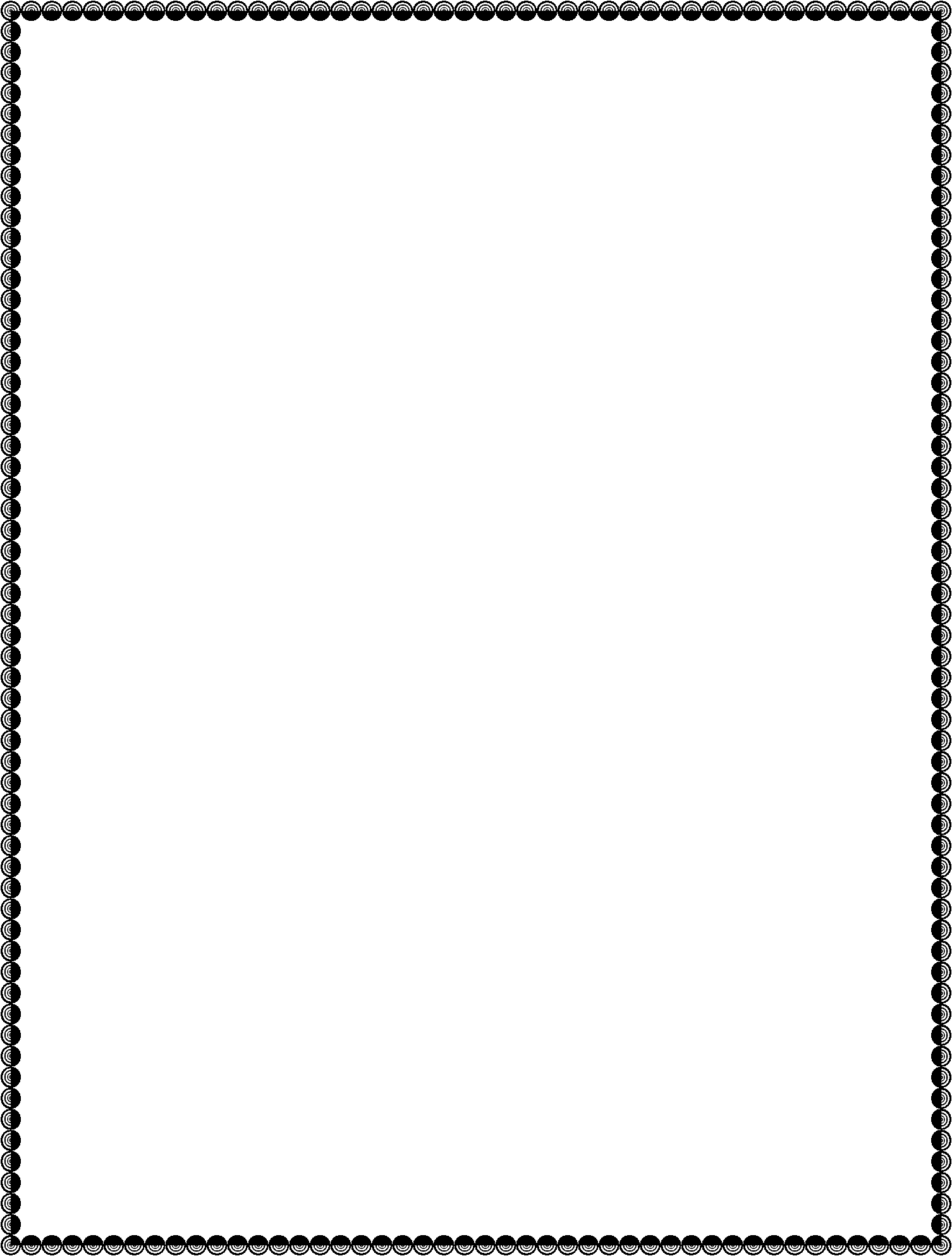
else{

cout << "\nPLAYER2 [O] named \"" << player\_o.name << "\" wins: " << player\_o.wins;

file<<"\nPLAYER2 [O] named \"" << player\_o.name << "\" wins: " << player\_o.wins;

}

cout << endl;



cout << "\nTie games: " << tie\_game << endl; file<< "\nTie games: " << tie\_game << endl; cout << endl;

printdesign(); cout << endl;

cout<<"\t\t Result: "<<endl;

file<<" ";

file<<" Result: ";

file<<" ";

printminus();

if (player\_x.wins > player\_o.wins) {

cout << "\nPLAYER1 [X] named: \"" << player\_x.name << "\" WINS THIS GAME!"; file<<"\nPLAYER1 [X] named: \"" << player\_x.name << "\" WINS THIS GAME!";

}

else if (player\_o.wins > player\_x.wins) { if(player\_x.user\_mode==2){

cout<<endl<< player\_o.name<<" [O] WINS THIS GAME!"; file<<endl<< player\_o.name<<" [O] WINS THIS GAME!";

}

else{

cout << "\nPLAYER2 [O] named: \"" << player\_o.name << "\" WINS THIS GAME!"; file<< "\nPLAYER2 [O] named: \"" << player\_o.name << "\" WINS THIS GAME!";

}

}

else if (player\_o.wins==player\_x.wins) {

cout << "\nIt is a tie!! no one wins the game!"; file<< "\nIt is a tie!! no one wins the game!";

}

}

}

void reset\_game(){

// Reset the board to its initial state with numbers from 1 to 9 char value='1';

for(int i=0;i<3;i++)

{

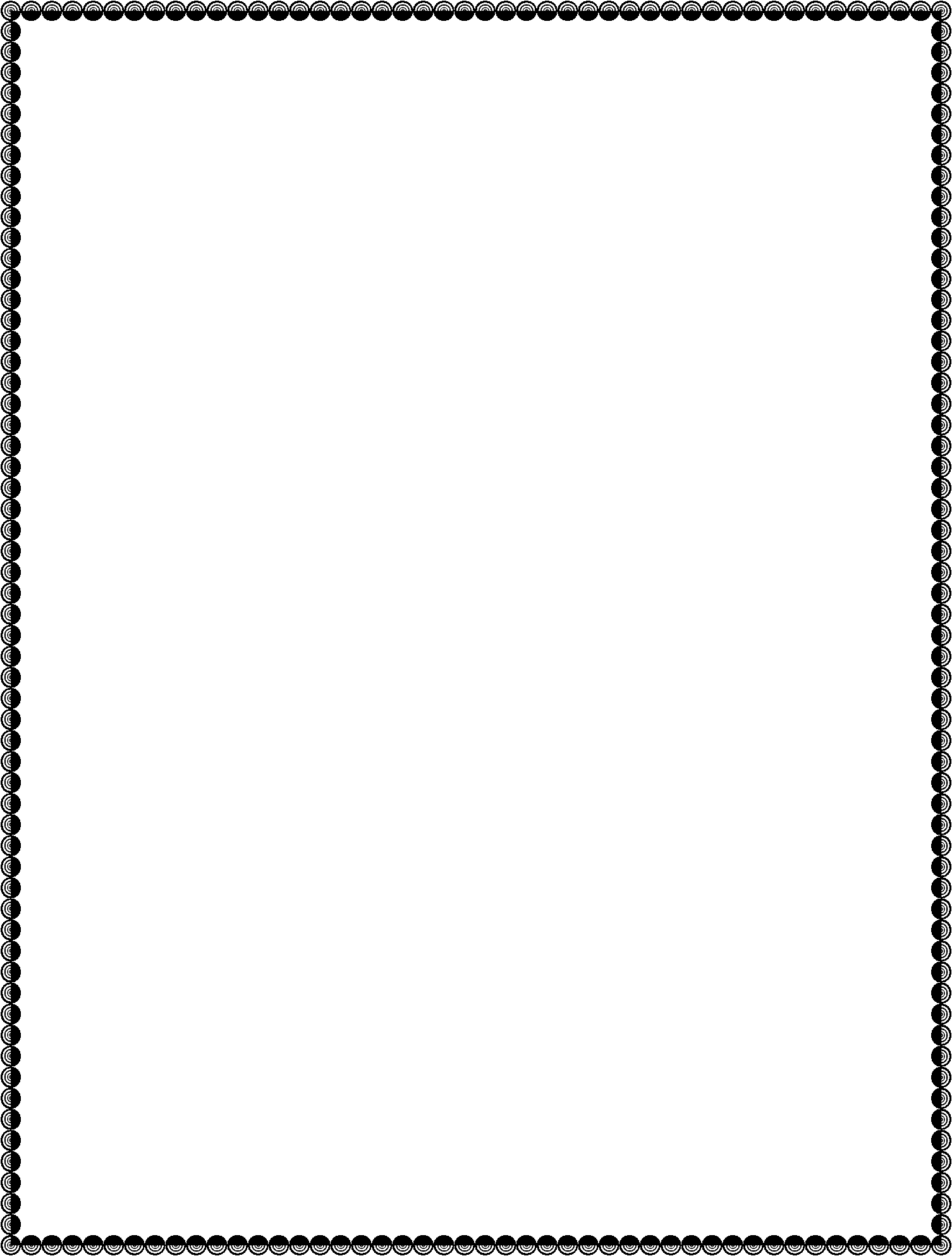
for(int k=0;k<3;k++){ board[i][k]=value; value++;

}

}

turn = 'X'; // Set the starting player to X

draw = false; // Reset the draw state



}

void playervsplayer(){ system("cls"); game\_label(); cout<<endl;

cout << "\n\t PLAYER1 [X] Vs PLAYER2 [O]\n\n"; cin.ignore();

cout <<endl<<"Enter your name Player[X]: "; getline(cin, player\_x.name); player\_x.symbol = 'X';

player\_x.wins = 0;

cout << "Enter your name Player[O]: "; getline(cin, player\_o.name); player\_o.symbol = 'O';

player\_o.wins = 0;

do { reset\_game();

while (game\_over()) { display\_board(); player\_turn();

}

if (turn == 'X' && !draw) {

cout << "Player2 [" << player\_o.symbol << "] named: \"" << player\_o.name << "\" WINS THIS GAME\nCONGRATULATIONS!\n";

player\_o.wins++;

} else if (turn == 'O' && !draw) {

cout << "Player1 [" << player\_x.symbol << "] named: \"" << player\_x.name << "\" WINS THIS GAME\nCONGRATULATIONS!\n";

player\_x.wins++;

} else {

cout << "IT'S A TIE!\n"; tie\_game++;

}

cout << "Do you want to play more?\nEnter (Y) for Yes and (N) for No :

";

cin >> user\_ask;

} while (user\_ask == 'Y' || user\_ask == 'y');

scoreboard();

}

void computervsplayer() { system("cls");

//showing label game\_label(); cout << endl;

cout << "\n\t Computer [O] Vs Player [X]\n\n"; cin.ignore();

cout << endl << "Enter your name Player[X]: "; getline(cin, player\_x.name);

player\_x.symbol = 'X'; player\_x.wins = 0; player\_o.name = "Computer"; player\_o.symbol = 'O'; player\_o.wins = 0;

do { reset\_game();

while (game\_over()) { if (turn == 'X') {

display\_board(); // Display the board before the human

player's turn

player\_turn(); // Human player's turn

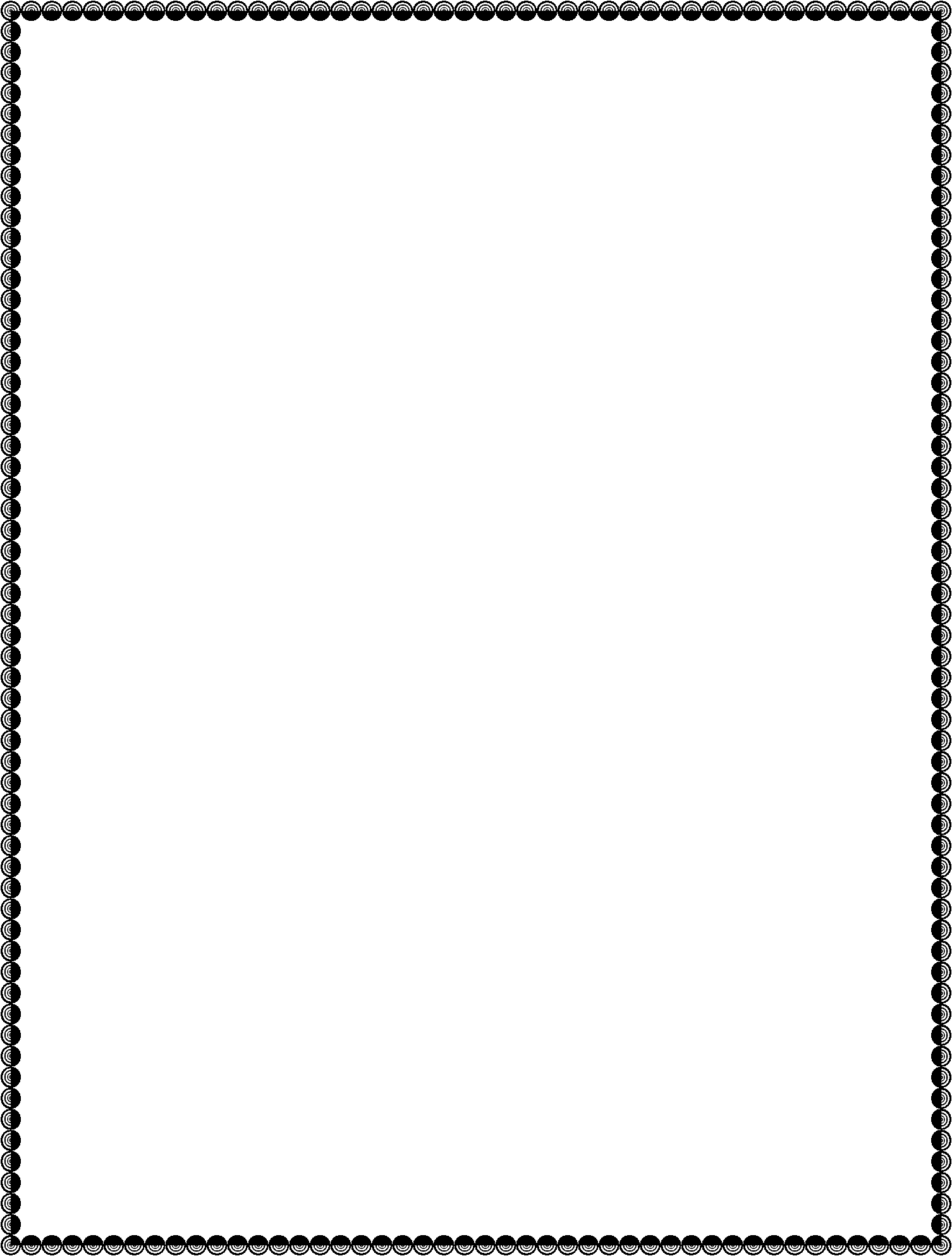
move

} else {

computer\_turn(); // Computer's turn

display\_board(); // Display the board after the computer's

}



}

if (turn == 'X' && !draw) {

cout << "Computer[" << player\_o.symbol << "] WINS THIS GAME\nCONGRATULATIONS!\n";

player\_o.wins++;

} else if (turn == 'O' && !draw) {

cout << "Player1 [" << player\_x.symbol << "] named: \"" << player\_x.name << "\" WINS THIS GAME\nCONGRATULATIONS!\n";

player\_x.wins++;

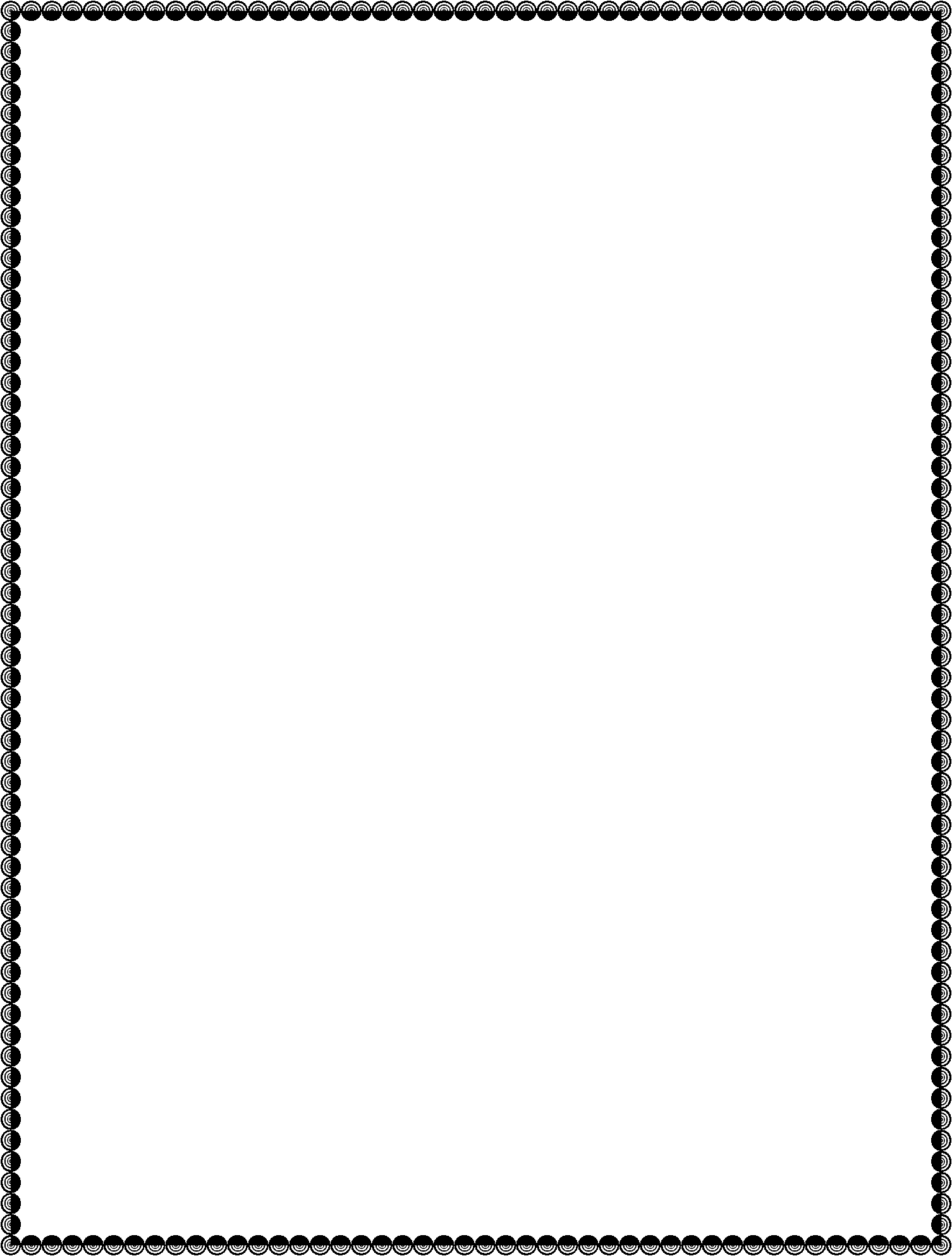
} else {

cout << "IT'S A TIE!\n"; tie\_game++;

}

cout << "Do you want to play more?\nEnter (Y) for Yes and (N) for No :

";



cin >> user\_ask;

} while (user\_ask == 'Y' || user\_ask == 'y');

scoreboard();

}

int main()

{

//asking for modes do {

game\_label();

cout << "\n\t\t Choose mode:\n"; printminus();

cout << "\n\t1.Player1 VS Player 2 (P1vsP2)\n"; cout << "\t2.Computer VS Player (CvsP)\n"; printminus();

cout << "\n\nEnter \"1\" for (P1vsP2) and \"2\" for (CvsP): "; cin >> player\_x.user\_mode;

if (player\_x.user\_mode == 1) { playervsplayer();

break;

} else if (player\_x.user\_mode == 2) { computervsplayer();

} else {

cout << "Invalid request for mode\n";

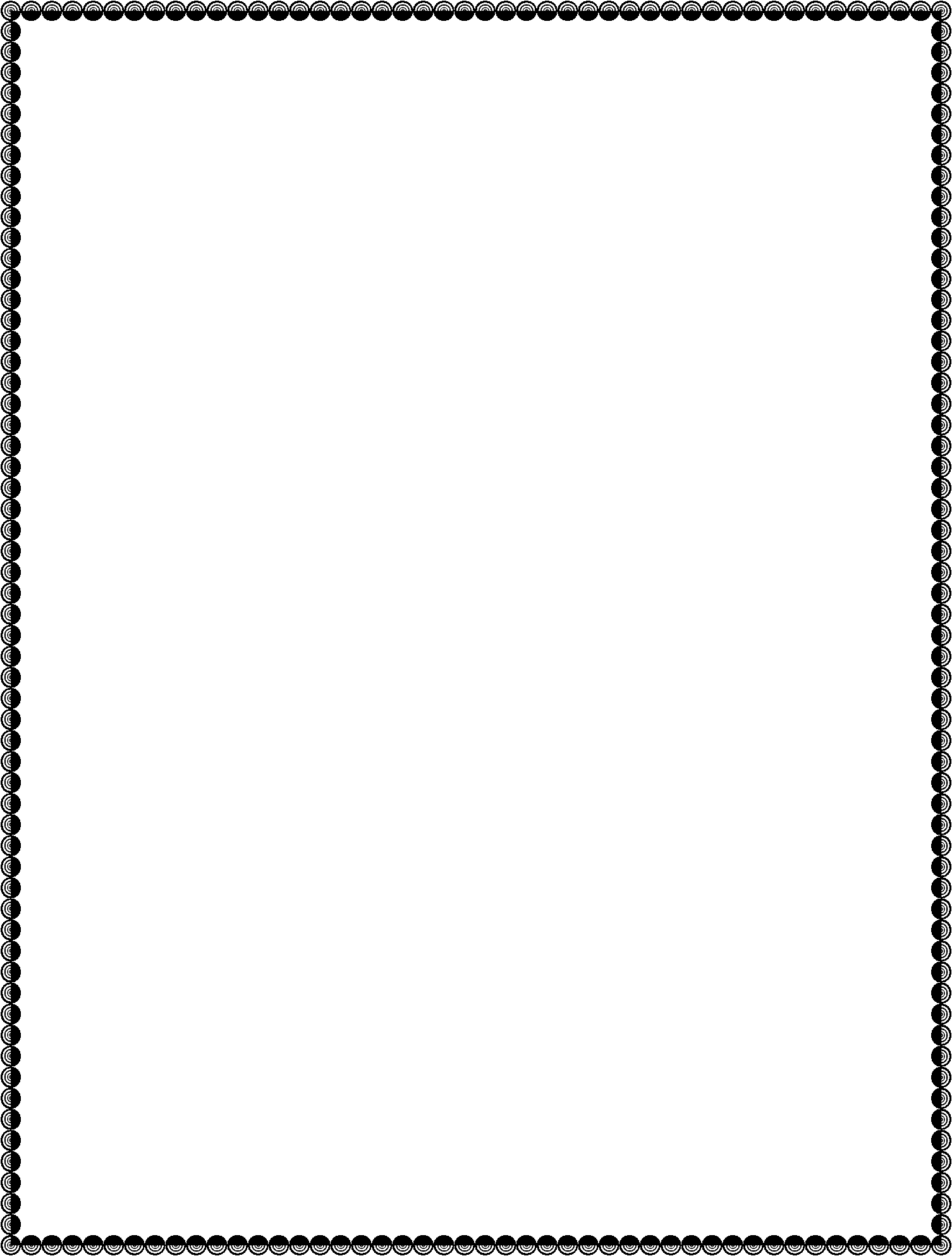
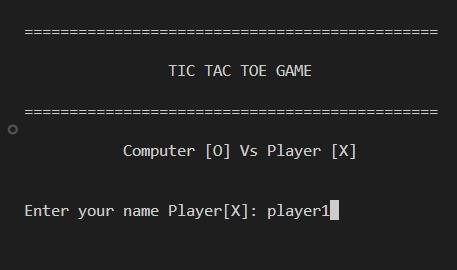
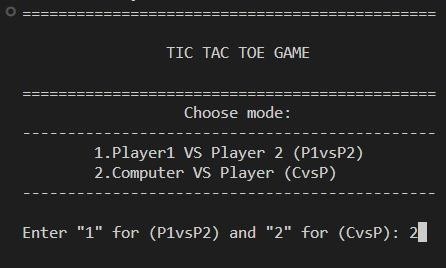
}

} while (player\_x.user\_mode != 1 && player\_x.user\_mode != 2);

return 0;

}

**OUTPUT**

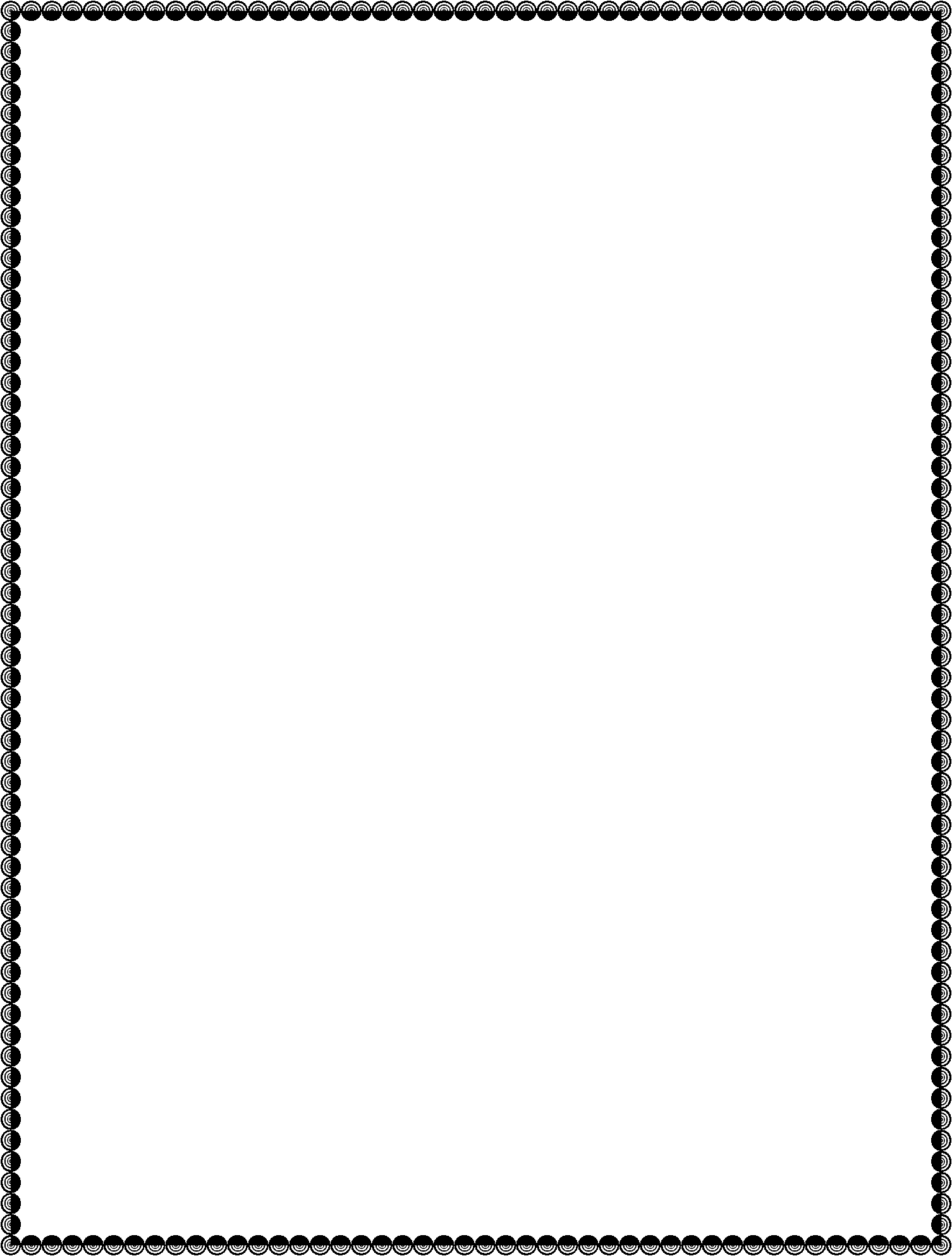


First when the user run the code then in console the game lable will shown and ask the user to enter the mode.

# COMPUTER VS PLAYER MODE:

If user enter 2 then the compiler will ask the user to enter the name.

When after this the player will press the enter, it clear the console and will say the player whose name is player 1 that it’s your turn. After every turn of the player the next turn is of computer.



Player 1 turn = 1 computer turn = 4;

Player 1 turn = 2 computer turn = 3;

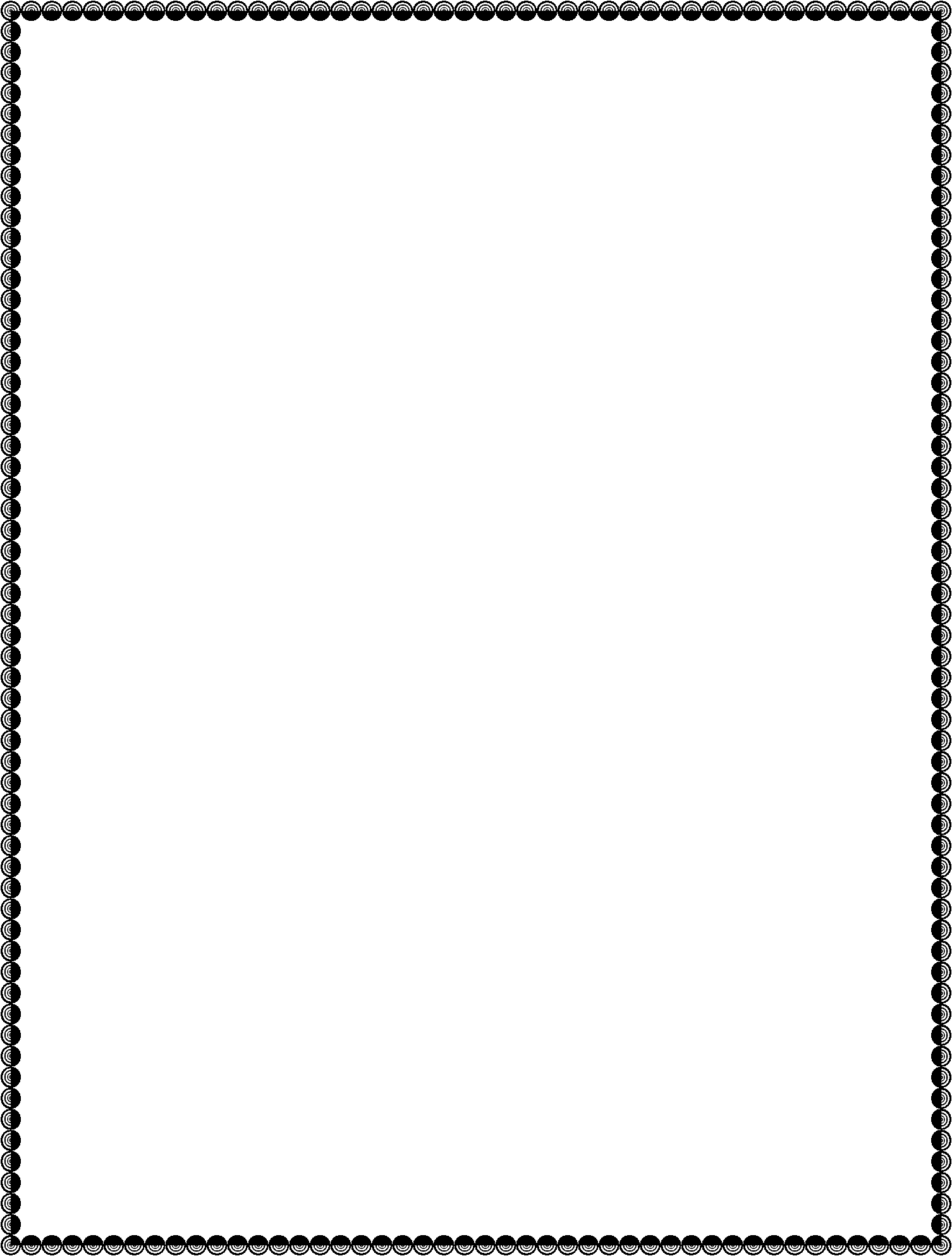
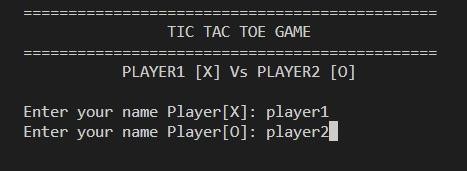
Player 1 turn = 5 computer turn = 8;

Player 1 turn = 6 computer turn = 9;

Player 1 turn = 7

Now if the player enter ‘n or N’ then it will clear the console and show the score board. If he enter ‘y or Y’ it will clear the console, board reset and again game will start.

# PLAYER VS PLAYER MODE:



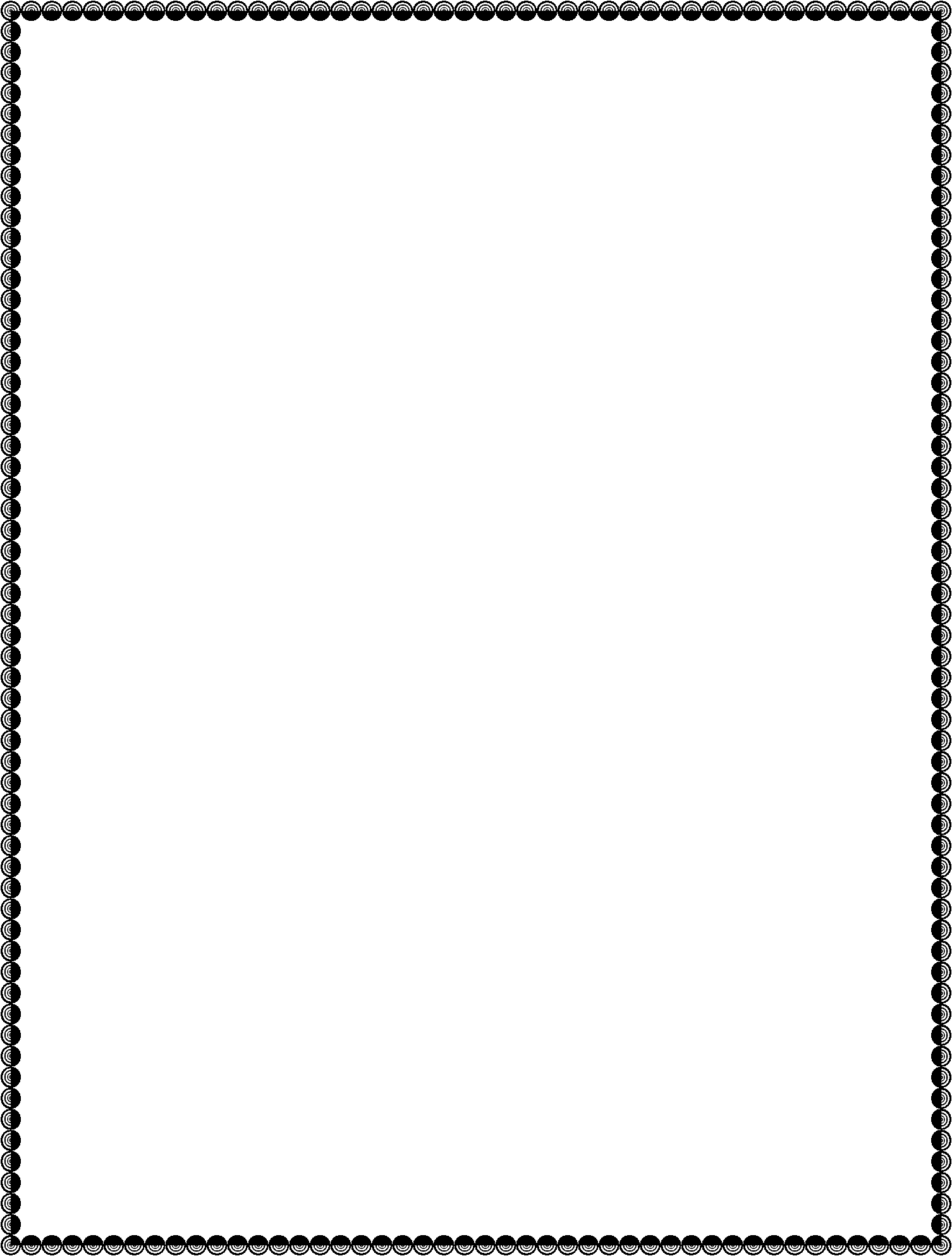
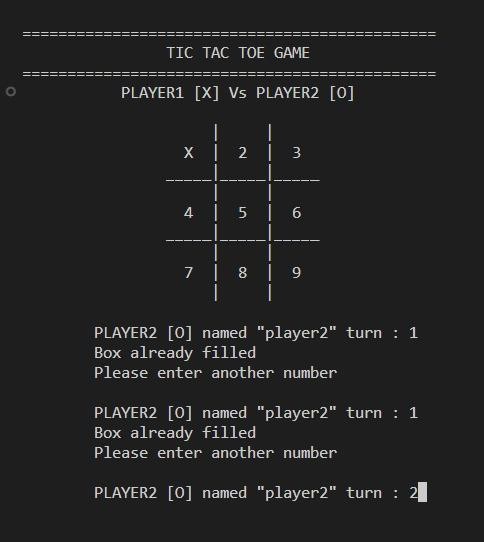
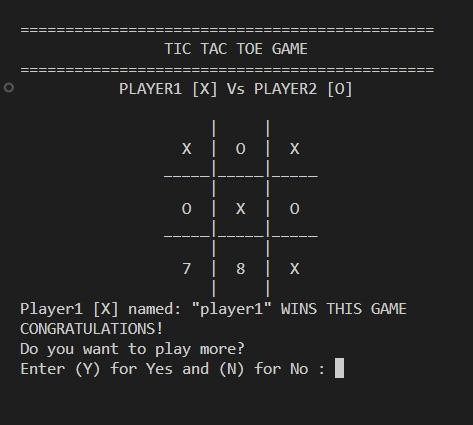
If user enter 2 then the compiler will ask the user to enter the name User enter the name of first player: player1

And the name of second player is: player 2

When after this the user will press the enter, it clear the console and will say the player1 that it’s your turn. Whenever the player enter his turn it again clear the console and change the number chosen by the user from 1 to 9 to its symbol [X] or [O]. And then give turn to the next player.

For example: player1 turn: ‘1’.

If the player will choose the same number then the message is shown by the compiler that the “box is already filled”. As long as player keeps selecting the same moves, the compiler will continue displaying the same message.



player 2 turn = 2 player 1 turn = 3;

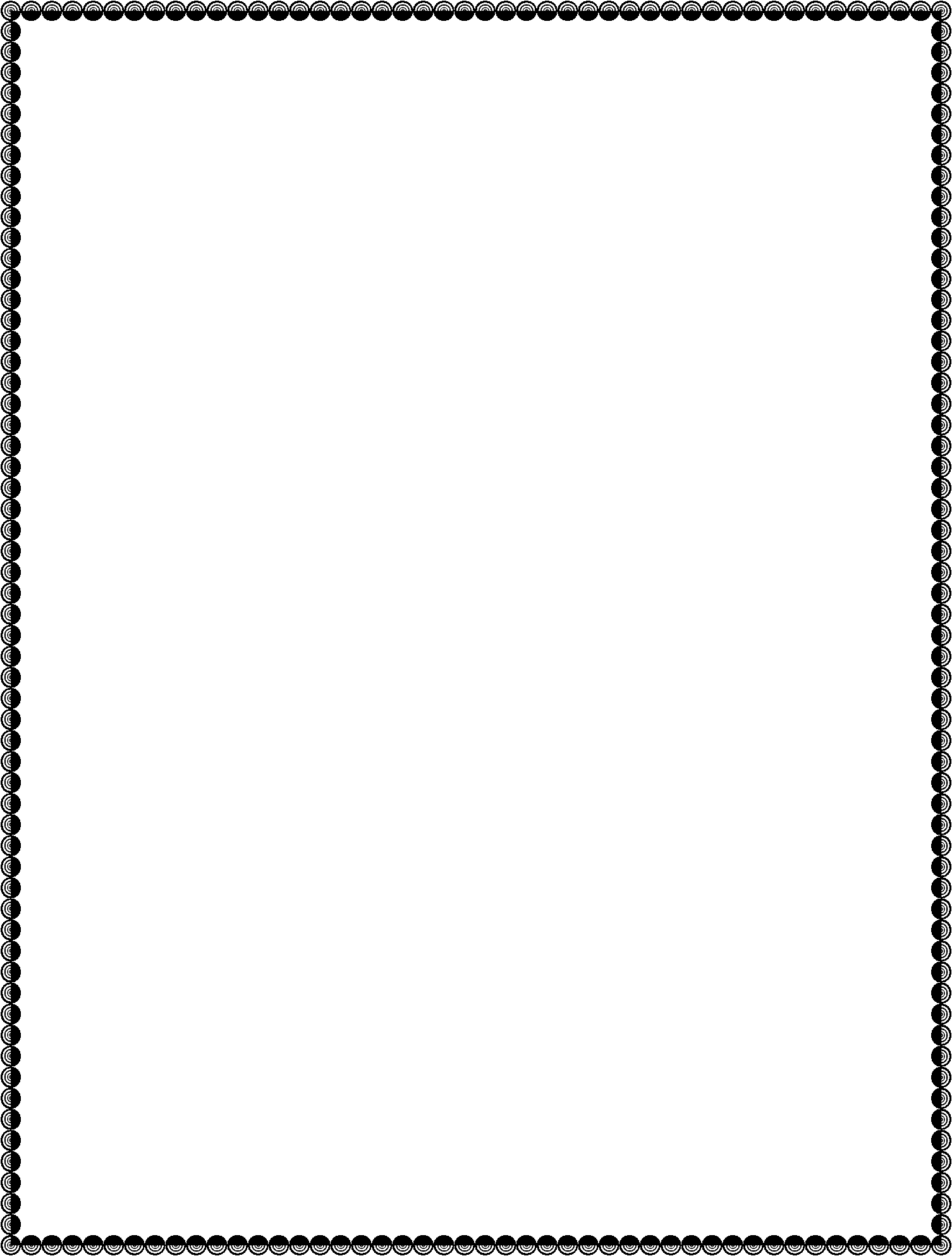
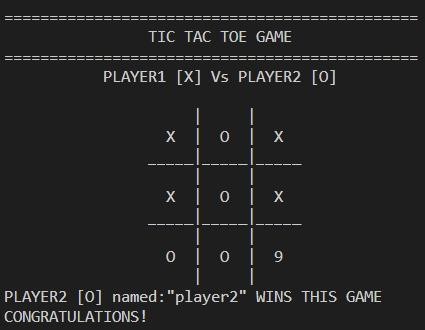
player 2 turn = 4 player 1 turn = 5;

player 2 turn = 6 player 1 turn = 9;

So, the player 1 wins this game. And then compiler will ask the player for further continue.

( Diagonal Win )

If player enter ‘y’ then it will clear the console ,board reset and again game will start.



Player 1 turn = 1 player 2 turn = 5;

Player 1 turn = 4 player 2 turn = 7;

Player 1 turn = 3 player 2 turn = 2;

Player 1 turn = 6 player 2 turn = 8;

So, the player 2 wins this game. And then compiler will ask the player for further continue.

( Verticale Win)

Again If player enter ‘y’ then it will clear the console, board reset and again game will start.

Player 1 turn = 5 player 2 turn = 7;

Player 1 turn = 1 player 2 turn = 9;

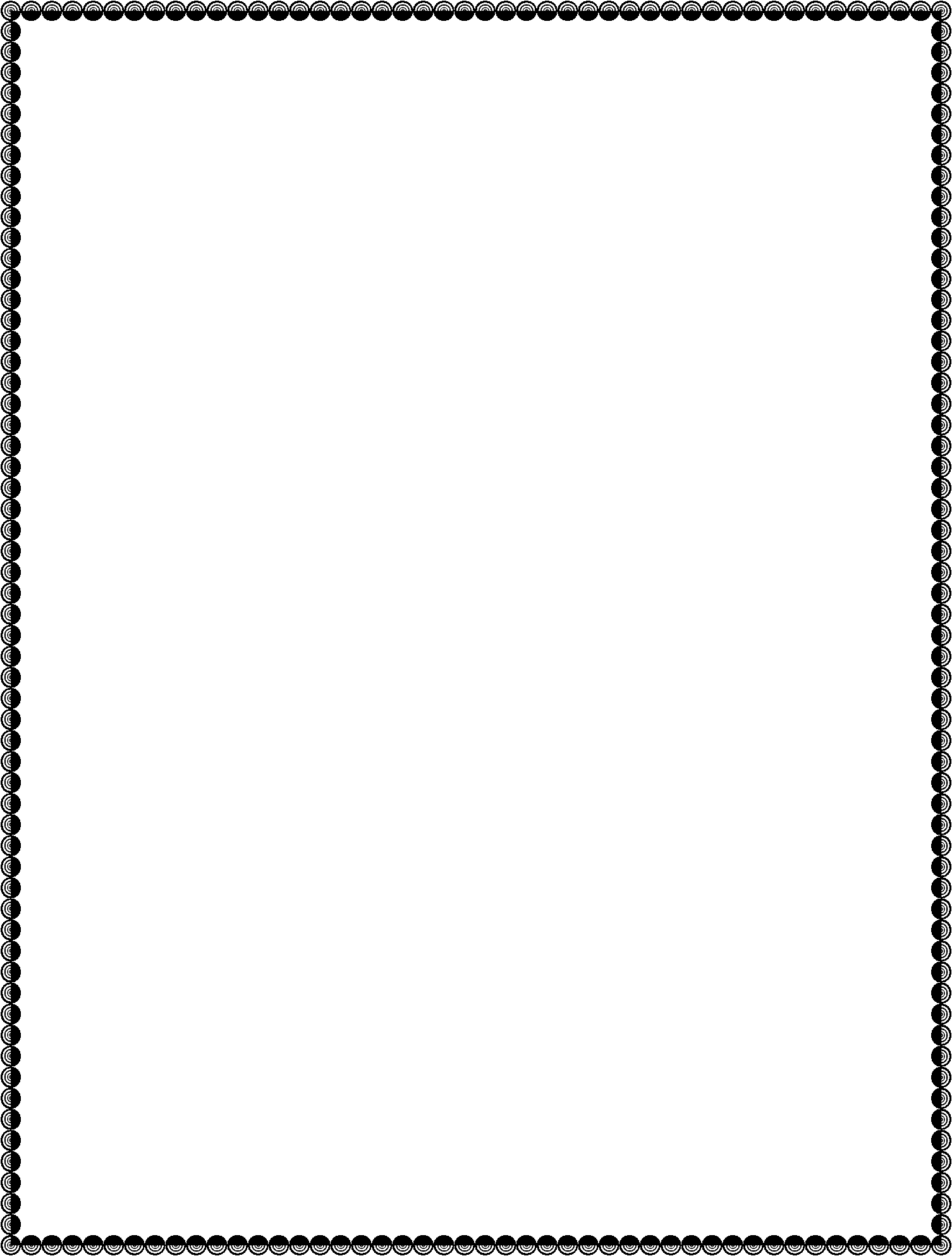
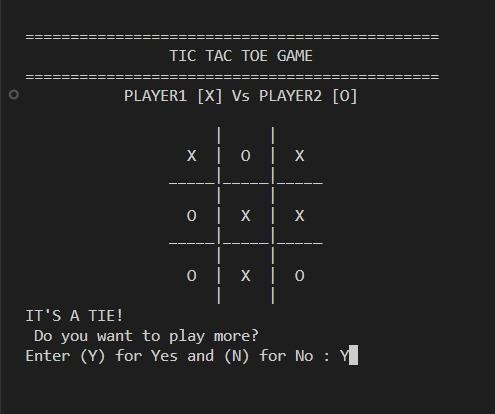
Player 1 turn =8 player 2 turn = 2;

Player 1 turn = 6 player 2 turn = 4;

Player 1 turn = 3

So, It’s a tie . And then compiler will ask the player for further continue.

( Tie Game )



If player enter ‘y’ then It will clear the console, board reset and again game will start.

Player 1 turn = 4 player 2 turn = 3;

Player 1 turn = 5 player 2 turn = 2;

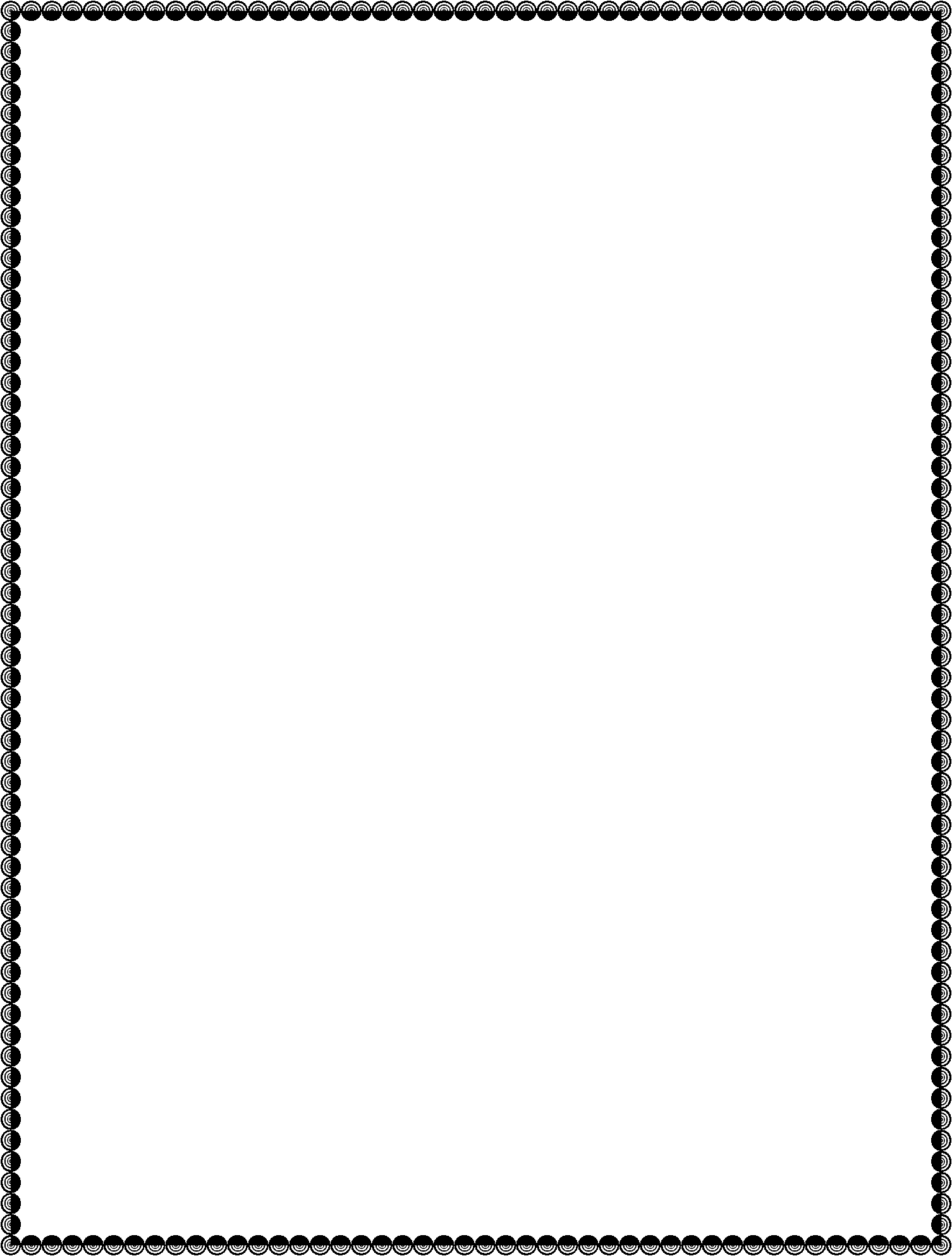
Player 1 turn = 1 player 2 turn = 7;

Player 1 turn = 6

So, the player 1 wins this game. And then compiler will ask the player for further continue.

( Horizontal Win)

Again If player enter ‘y’ then it will clear the console, board reset and again game will start.



Player 1 turn = 1 player 2 turn = 2;

Player 1 turn = 4 player 2 turn = 5;

Player 1 turn =3 player 2 turn = 7;

Player 1 turn = 8 player 2 turn = 6;

Player 1 turn = 9

So, It’s a tie game. And then compiler will ask the player for further continue. ( Tie Game )

If player enter ‘ n or N ’ then it will clear the console and show the scoreboard and the name of the winner.

