#include<iostream>

#include<stdlib.h>

#include <ctime>

using namespace std ;

bool gameOver = false;

//Function Prototypes

void cellnumbers();

void showtable (char TICTAC[][3]);

int input\_r\_c(int &row, int &column);

void player\_1(int &rows,int &columns,char TICTAC[][3], string &p1\_name, char &winner );

void computer(char TICTAC[][3], char &winner, string &p1\_name ,int &rows,int &columns );

char check\_winner\_row(char TICTAC[][3], char &Winner );

char check\_winner\_column(char TICTAC[][3], char &Winner);

char check\_winner\_diagonal(char TICTAC[][3], char &Winner);

void who\_won(char &winner, string &p1\_name);

char draw (char TICTAC[][3], char winner, int &row , int &column , string &p1\_name );

void playagain(char TICTAC[][3] , char &winner, string &p1\_name ,int &row,int &column);

int main ()

{

cout<<endl<<" Tic Tac Toe Game"<<endl<<endl;

cout<<" Please enter your name : ";

string p1\_name ;

cin>>p1\_name;

cout<<" " << p1\_name <<" : ( X ) VS Computer : ( O ) " <<endl;

char TICTAC[3][3]= { {}, {}, {} }, winner='t';

int row, column;

cellnumbers();

while(!gameOver)

{

player\_1( row, column, TICTAC, p1\_name, winner );

draw ( TICTAC, winner , row, column, p1\_name );

computer( TICTAC, winner, p1\_name , row , column );

draw ( TICTAC, winner, row, column, p1\_name );

}

who\_won(winner, p1\_name);

while(1)

{

playagain(TICTAC,winner,p1\_name ,row,column);

}

}

//User Defined Functions

//Function to show how the table cells are numbered

void cellnumbers()

{

cout<<" please enter the rows and columns according to this figure : "<<endl;

cout <<" | | "<<endl

<<" 1,1 | 1,2 | 1,3 "<<endl

<<" \_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_"<<endl

<<" | | "<<endl

<<" 2,1 | 2,2 | 2,3 "<<endl

<<" \_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_"<<endl

<<" | | "<<endl

<<" 3,1 | 3,2 | 3,3 "<<endl

<<" | | "<<endl;

}

//Function to input the desired position of the player and return the row and column values

int input\_r\_c (int &row, int &column)

{

cin>>row>>column;

if(row<1 || row>3 || column<1 || column>3)

{

cout<<"Invalid , Please enter correct row and column again" <<endl ;

input\_r\_c(row, column) ;

}

return row-1;

return column-1;

}

//Function to print the tictactoe table after inserting the user's move

void showtable ( char TICTAC[][3])

{

system("cls");

cout<<" | | " <<endl

<<" "<<TICTAC[0][0]<<" | "<<TICTAC[0][1]<<" | "<<TICTAC[0][2]<<endl

<<" \_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" <<endl

<<" | | " <<endl

<<" "<<TICTAC[1][0]<<" | "<<TICTAC[1][1]<<" | "<<TICTAC[1][2]<<endl

<<" \_\_\_\_\_|\_\_\_\_\_|\_\_\_\_\_" <<endl

<<" | | " <<endl

<<" "<<TICTAC[2][0]<<" | "<<TICTAC[2][1]<<" | "<<TICTAC[2][2]<<endl

<<" | | " <<endl;

}

//Functions to check if there is a winner

void player\_1(int &row,int &column,char TICTAC[][3],string &p1\_name, char &winner )

{

cout<<" "<<p1\_name<<" , please choose your position and 'X' will be inserted "<<endl;

input\_r\_c (row, column);

if(TICTAC[row-1][column-1]== 'X' || TICTAC[row-1][column-1]== 'O')

{

cout<<" Invalid move, try again.\n";

cout<<" "<<p1\_name<<" , please choose your position and 'X' will be inserted "<<endl;

input\_r\_c (row, column);

draw ( TICTAC, winner , row, column, p1\_name );

TICTAC[row-1][column-1]='X';

check\_winner\_row(TICTAC, winner );

check\_winner\_column(TICTAC, winner);

check\_winner\_diagonal(TICTAC, winner );

}

draw ( TICTAC, winner , row, column, p1\_name );

TICTAC[row-1][column-1]='X';

check\_winner\_row(TICTAC, winner );

check\_winner\_column(TICTAC, winner);

check\_winner\_diagonal(TICTAC, winner );

showtable ( TICTAC );

}

//computer's turn

void computer(char TICTAC[][3], char &winner, string &p1\_name , int &row , int &column)

{

check :

srand(time(0));

int Move = rand() % 3 +1;

int movee = rand() % 3 + 1;

if(TICTAC[Move-1][movee-1]!='X' && TICTAC[Move-1][movee-1]!='O' && gameOver==false)

{

cout << " Computer Move : "<< Move <<" " << movee <<endl;

TICTAC[Move-1][movee-1]='O';

}

else if (TICTAC[Move-1][movee-1]=='X' || TICTAC[Move-1][movee-1]=='O' )

{

goto check ;

}

draw ( TICTAC, winner , row, column, p1\_name );

check\_winner\_row(TICTAC, winner );

check\_winner\_column(TICTAC, winner);

check\_winner\_diagonal(TICTAC, winner );

showtable ( TICTAC );

}

//checking if any of the rows is the same

char check\_winner\_row(char TICTAC[][3], char &winner )

{

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

{

if( TICTAC[i][0]==TICTAC[i][1] && TICTAC[i][1]==TICTAC[i][2])

{

if(TICTAC[i][0]=='X' || TICTAC[i][0]=='O' )

{

winner = TICTAC[i][0];

gameOver=true ;

return winner ;

exit(0);

}

}

}

}

//checking if any of the columns is the same

char check\_winner\_column(char TICTAC[][3], char &winner )

{

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

{

if( TICTAC[0][i]==TICTAC[1][i] && TICTAC[1][i]==TICTAC[2][i])

{

if( TICTAC[0][i]=='X' || TICTAC[0][i]=='O' )

{

winner = TICTAC[0][i];

gameOver=true ;

return winner;

exit(0);

}

}

}

}

//checking both diagonals

char check\_winner\_diagonal(char TICTAC[][3], char &winner )

{

if( TICTAC[0][0]==TICTAC[1][1] && TICTAC[1][1]==TICTAC[2][2] )

{

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

{

winner = TICTAC[0][0];

gameOver=true ;

return winner ;

exit(0);

}

}

if( TICTAC[0][2]==TICTAC[1][1] && TICTAC[1][1]==TICTAC[2][0])

{

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

{

winner = TICTAC[0][2];

gameOver=true ;

return winner ;

exit(0);

}

}

}

//function to check who won the game

void who\_won (char &winner, string & p1\_name )

{

if(winner == 'X')

cout<<" \a Game Over ! \n You won the game , Congratulations ! :)"<<endl;

else if(winner == 'O')

cout<<" \a Game Over ! " <<" \n Computer won the game :) "<<endl;

else if(winner == 't')

cout<<" \a Game Over ! \n TIE :( "<<endl ;

}

//function to check for ties

char draw (char TICTAC[][3], char winner , int &row , int &column , string &p1\_name)

{

int counter=0;

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

{

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

{

if(TICTAC[i][j]=='X' || TICTAC[i][j]=='O')

counter+=1;

}

}

if ( counter == 9 )

{

check\_winner\_row(TICTAC, winner );

check\_winner\_column(TICTAC, winner);

check\_winner\_diagonal(TICTAC, winner );

if(winner=='O' || winner=='X')

{

who\_won(winner, p1\_name) ;

playagain(TICTAC,winner,p1\_name ,row,column);

}

if(winner!='O' && winner!='X')

{

winner = 't';

gameOver = true ;

cout<<" \a Game Over \n TIE :( \n ";

playagain(TICTAC,winner,p1\_name ,row,column);

}

}

}

//function gives an option to play again

void playagain(char TICTAC[][3] , char &winner, string &p1\_name ,int &row,int &column)

{

char option;

cout << "Wanna play again? Y for yes, N for no" << endl;

cin >> option;

while(option != 'Y' && option != 'y' && option != 'N' && option != 'n')

{

cout << "invalid input enter again" << endl;

cin >> option;

}

if (option == 'Y' || option == 'y')

{

gameOver = false ;

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

{

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

{

if(TICTAC[i][j]=='X' || TICTAC[i][j]=='O');

TICTAC[i][j] = ' ';

}

}

winner='t';

goto playagain ;

}

else if(option == 'N' || option == 'n')

{

exit(0);

}

playagain :

cellnumbers();

while(!gameOver)

{

player\_1( row, column, TICTAC, p1\_name, winner );

draw ( TICTAC, winner, row, column, p1\_name );

computer( TICTAC, winner, p1\_name ,row , column );

draw ( TICTAC, winner, row, column, p1\_name );

}

who\_won(winner, p1\_name);

}