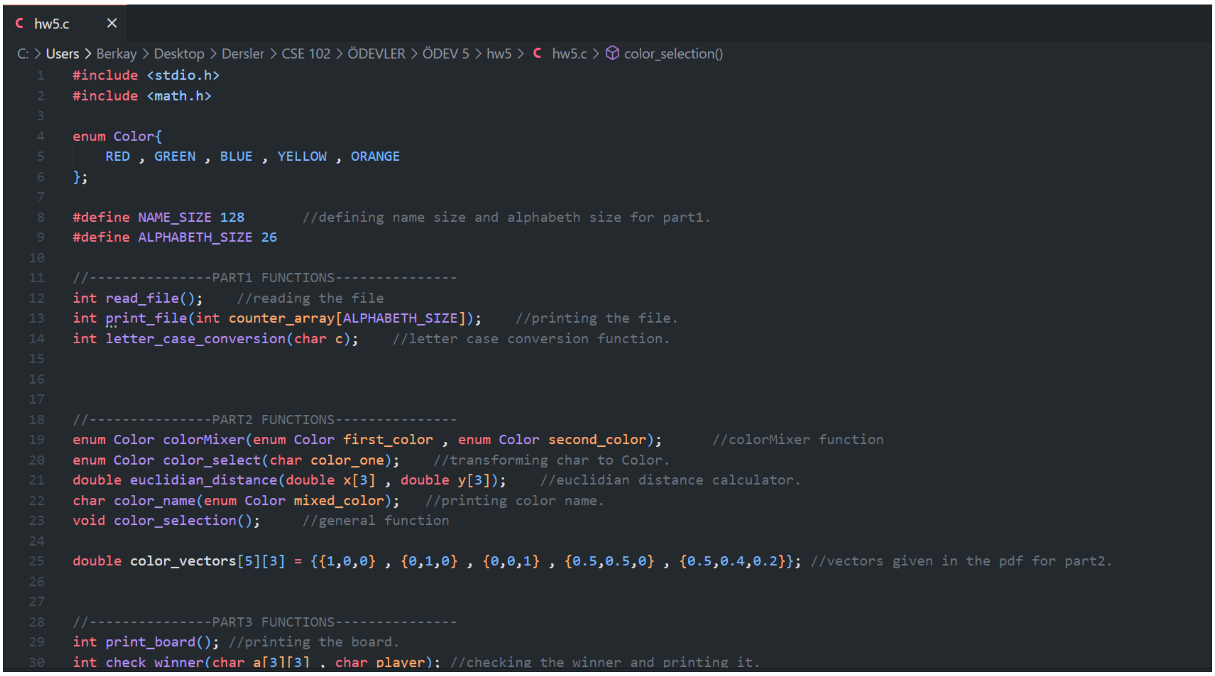
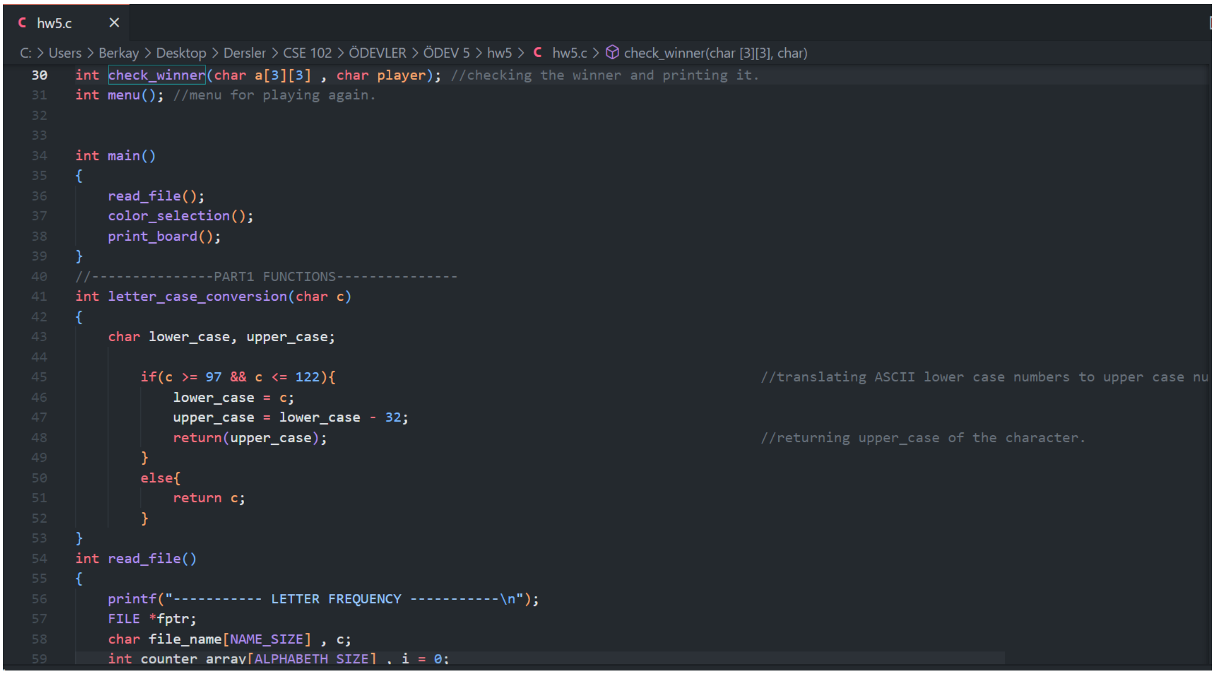
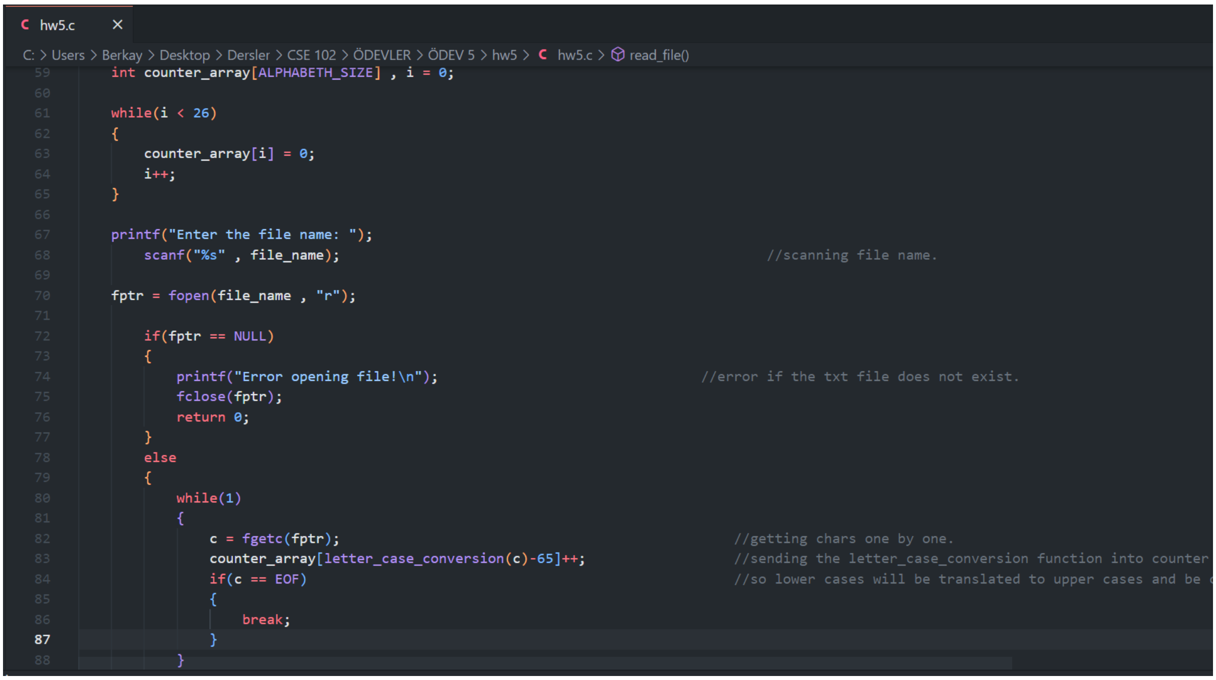

hw5.c ×
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > C hw5.c > color_selection()

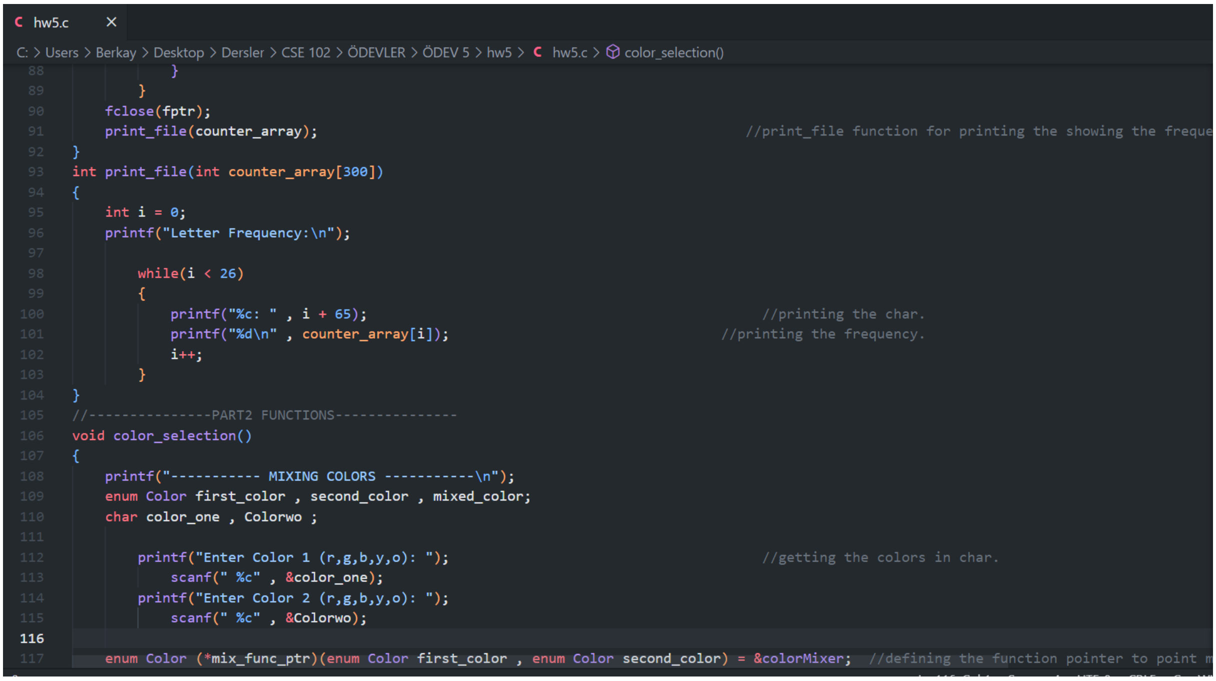
```c
#include <stdio.h>
#include <math.h>

enum Color{
    RED , GREEN , BLUE , YELLOW , ORANGE
};

#define NAME_SIZE 128       //defining name size and alphabeth size for part1.
#define ALPHABETH_SIZE 26

//---------------PART1 FUNCTIONS---------------
int read_file();    //reading the file
int print_file(int counter_array[ALPHABETH_SIZE]);    //printing the file.
int letter_case_conversion(char c);    //letter case conversion function.



//---------------PART2 FUNCTIONS---------------
enum Color colorMixer(enum Color first_color , enum Color second_color);      //colorMixer function
enum Color color_select(char color_one);    //transforming char to Color.
double euclidian_distance(double x[3] , double y[3]);    //euclidian distance calculator.
char color_name(enum Color mixed_color);    //printing color name.
void color_selection();      //general function

double color_vectors[5][3] = {{1,0,0} , {0,1,0} , {0,0,1} , {0.5,0.5,0} , {0.5,0.4,0.2}}; //vectors given in the pdf for part2.


//---------------PART3 FUNCTIONS---------------
int print_board(); //printing the board.
int check_winner(char a[3][3] , char player); //checking the winner and printing it.
```


hw5.c ×
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > C hw5.c > check_winner(char [3][3], char)

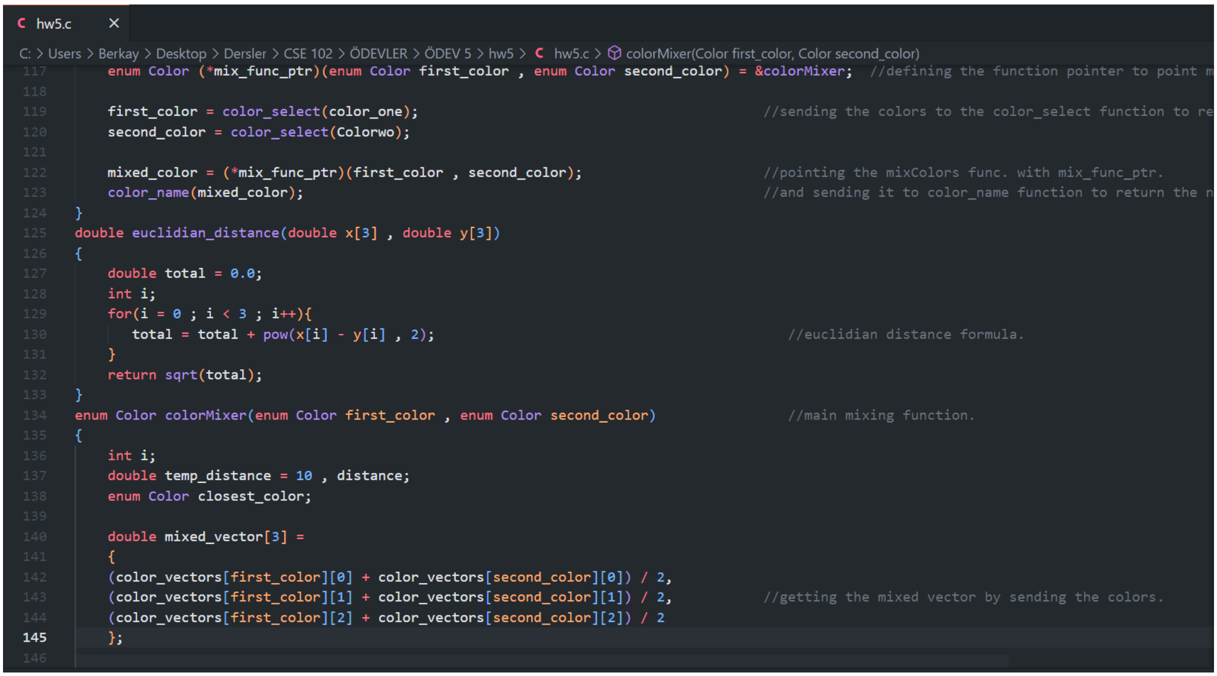
```c
int check_winner(char a[3][3] , char player); //checking the winner and printing it.
int menu(); //menu for playing again.


int main()
{
    read_file();
    color_selection();
    print_board();
}
//---------------PART1 FUNCTIONS---------------
int letter_case_conversion(char c)
{
    char lower_case, upper_case;

        if(c >= 97 && c <= 122){                                            //translating ASCII lower case numbers to upper case nu
            lower_case = c;
            upper_case = lower_case - 32;
            return(upper_case);                                             //returning upper_case of the character.
        }
        else{
            return c;
        }
}
int read_file()
{
    printf("----------- LETTER FREQUENCY -----------\n");
    FILE *fptr;
    char file_name[NAME_SIZE] , c;
    int counter_array[ALPHABETH_SIZE] , i = 0;
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > read_file()

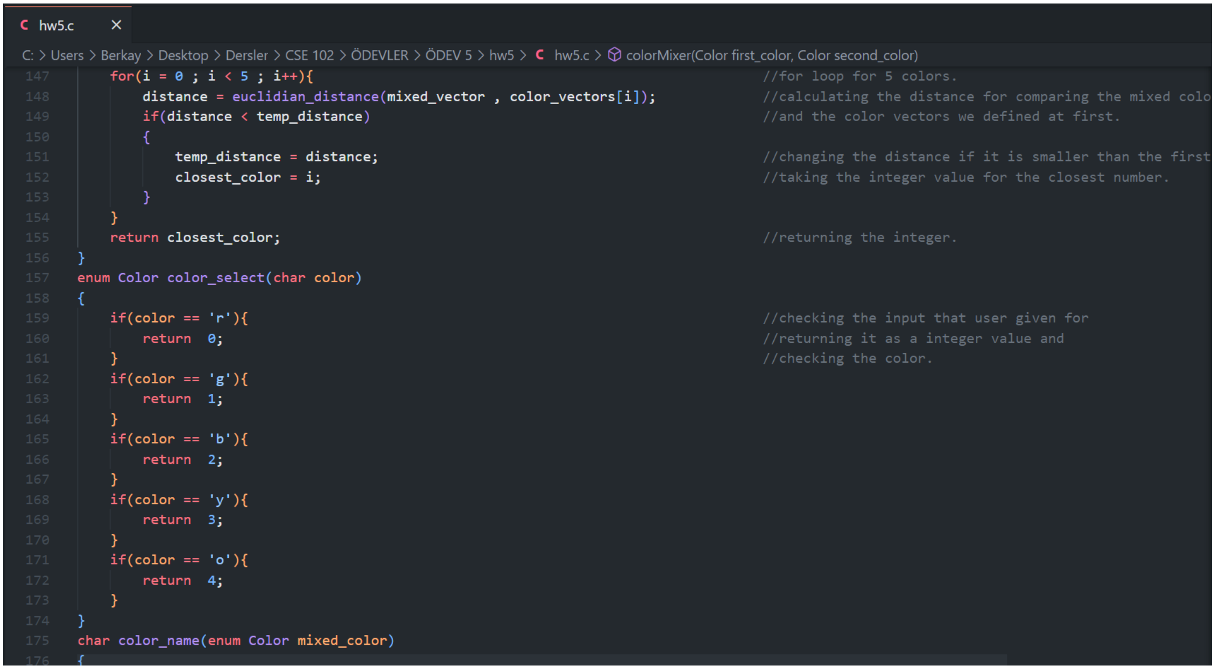
```c
    int counter_array[ALPHABETH_SIZE] , i = 0;

    while(i < 26)
    {
        counter_array[i] = 0;
        i++;
    }

    printf("Enter the file name: ");
        scanf("%s" , file_name);                                                    //scanning file name.

    fptr = fopen(file_name , "r");

        if(fptr == NULL)
        {
            printf("Error opening file!\n");                                //error if the txt file does not exist.
            fclose(fptr);
            return 0;
        }
        else
        {
            while(1)
            {
                c = fgetc(fptr);                                                //getting chars one by one.
                counter_array[letter_case_conversion(c)-65]++;                  //sending the letter_case_conversion function into counter
                if(c == EOF)                                                    //so lower cases will be translated to upper cases and be 
                {
                    break;
                }
            }
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > color_selection()

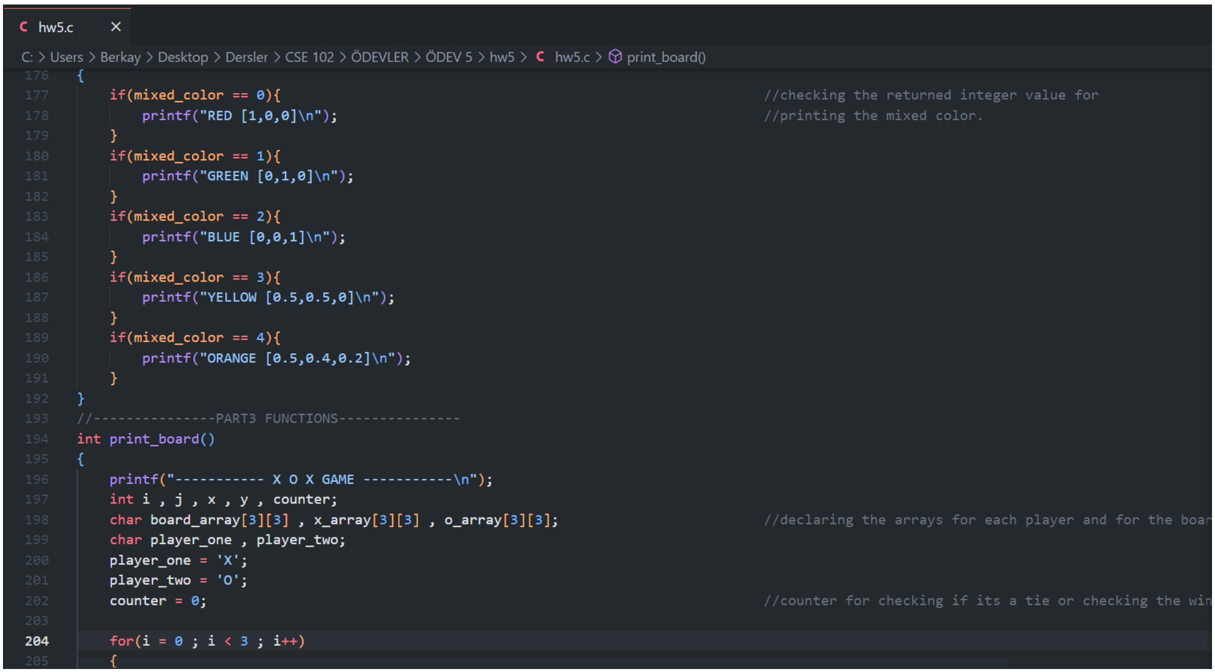
```c
            }
        }
    fclose(fptr);
    print_file(counter_array);                                                        //print_file function for printing the showing the freque
}
int print_file(int counter_array[300])
{
    int i = 0;
    printf("Letter Frequency:\n");

        while(i < 26)
        {
            printf("%c: " , i + 65);                                                    //printing the char.
            printf("%d\n" , counter_array[i]);                                     //printing the frequency.
            i++;
        }
}
//--------------PART2 FUNCTIONS--------------
void color_selection()
{
    printf("---------- MIXING COLORS ----------\n");
    enum Color first_color , second_color , mixed_color;
    char color_one , Colorwo ;

        printf("Enter Color 1 (r,g,b,y,o): ");                                          //getting the colors in char.
            scanf(" %c" , &color_one);
        printf("Enter Color 2 (r,g,b,y,o): ");
            scanf(" %c" , &Colorwo);

    enum Color (*mix_func_ptr)(enum Color first_color , enum Color second_color) = &colorMixer;  //defining the function pointer to point m
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > colorMixer(Color first_color, Color second_color)

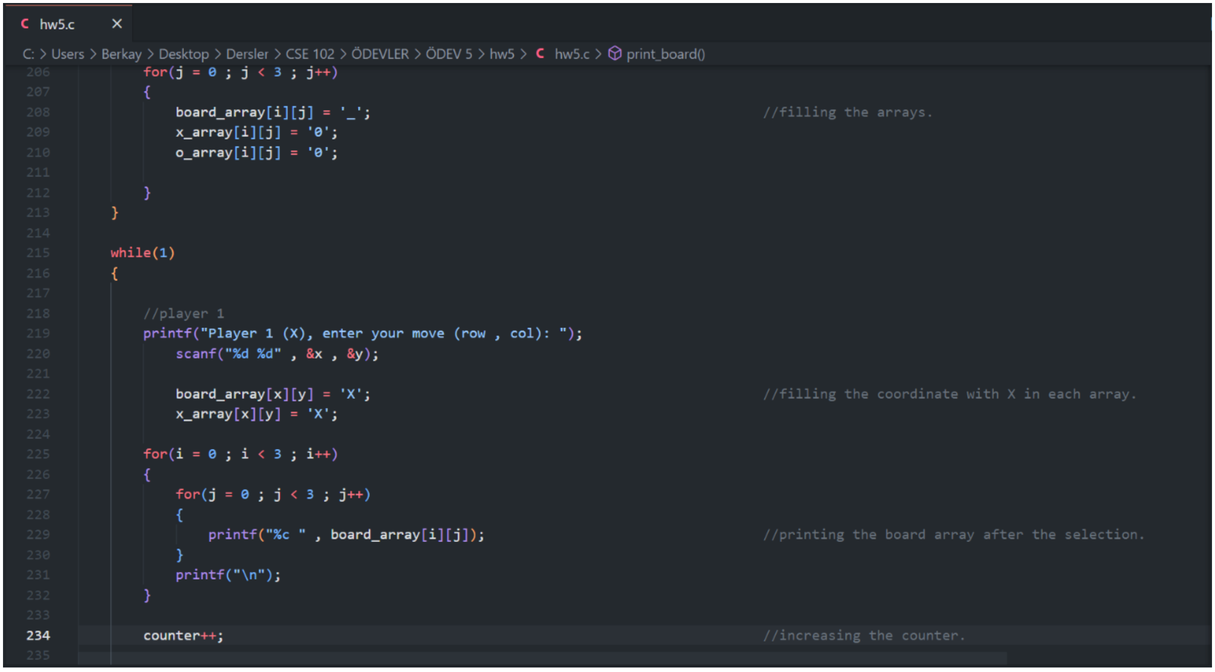
```c
    enum Color (*mix_func_ptr)(enum Color first_color , enum Color second_color) = &colorMixer;  //defining the function pointer to point m

    first_color = color_select(color_one);                                          //sending the colors to the color_select function to re
    second_color = color_select(Colorwo);

    mixed_color = (*mix_func_ptr)(first_color , second_color);                      //pointing the mixColors func. with mix_func_ptr.
    color_name(mixed_color);                                                        //and sending it to color_name function to return the n
}
double euclidian_distance(double x[3] , double y[3])
{
    double total = 0.0;
    int i;
    for(i = 0 ; i < 3 ; i++){
       total = total + pow(x[i] - y[i] , 2);                                           //euclidian distance formula.
    }
    return sqrt(total);
}
enum Color colorMixer(enum Color first_color , enum Color second_color)                //main mixing function.
{
    int i;
    double temp_distance = 10 , distance;
    enum Color closest_color;

    double mixed_vector[3] =
    {
    (color_vectors[first_color][0] + color_vectors[second_color][0]) / 2,
    (color_vectors[first_color][1] + color_vectors[second_color][1]) / 2,           //getting the mixed vector by sending the colors.
    (color_vectors[first_color][2] + color_vectors[second_color][2]) / 2
    };

```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > colorMixer(Color first_color, Color second_color)

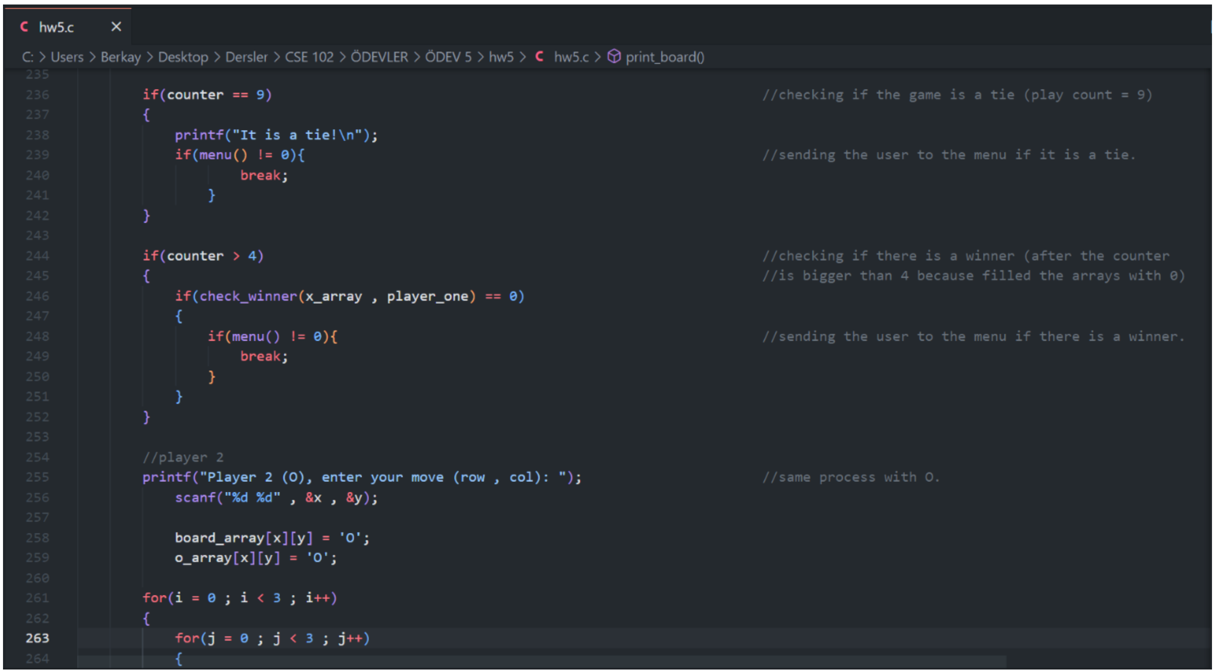
```c
    for(i = 0 ; i < 5 ; i++){                                                       //for loop for 5 colors.
        distance = euclidian_distance(mixed_vector , color_vectors[i]);             //calculating the distance for comparing the mixed colo
        if(distance < temp_distance)                                                //and the color vectors we defined at first.
        {
            temp_distance = distance;                                               //changing the distance if it is smaller than the first
            closest_color = i;                                                      //taking the integer value for the closest number.
        }
    }
    return closest_color;                                                           //returning the integer.
}
enum Color color_select(char color)
{
    if(color == 'r'){                                                               //checking the input that user given for
        return  0;                                                                  //returning it as a integer value and
    }                                                                               //checking the color.
    if(color == 'g'){
        return  1;
    }
    if(color == 'b'){
        return  2;
    }
    if(color == 'y'){
        return  3;
    }
    if(color == 'o'){
        return  4;
    }
}
char color_name(enum Color mixed_color)
{
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

```c
{
    if(mixed_color == 0){                                                           //checking the returned integer value for
        printf("RED [1,0,0]\n");                                                    //printing the mixed color.
    }
    if(mixed_color == 1){
        printf("GREEN [0,1,0]\n");
    }
    if(mixed_color == 2){
        printf("BLUE [0,0,1]\n");
    }
    if(mixed_color == 3){
        printf("YELLOW [0.5,0.5,0]\n");
    }
    if(mixed_color == 4){
        printf("ORANGE [0.5,0.4,0.2]\n");
    }
}
//--------------PART3 FUNCTIONS--------------
int print_board()
{
    printf("----------- X O X GAME -----------\n");
    int i , j , x , y , counter;
    char board_array[3][3] , x_array[3][3] , o_array[3][3];                         //declaring the arrays for each player and for the boar
    char player_one , player_two;
    player_one = 'X';
    player_two = 'O';
    counter = 0;                                                                    //counter for checking if its a tie or checking the win

    for(i = 0 ; i < 3 ; i++)
    {
```

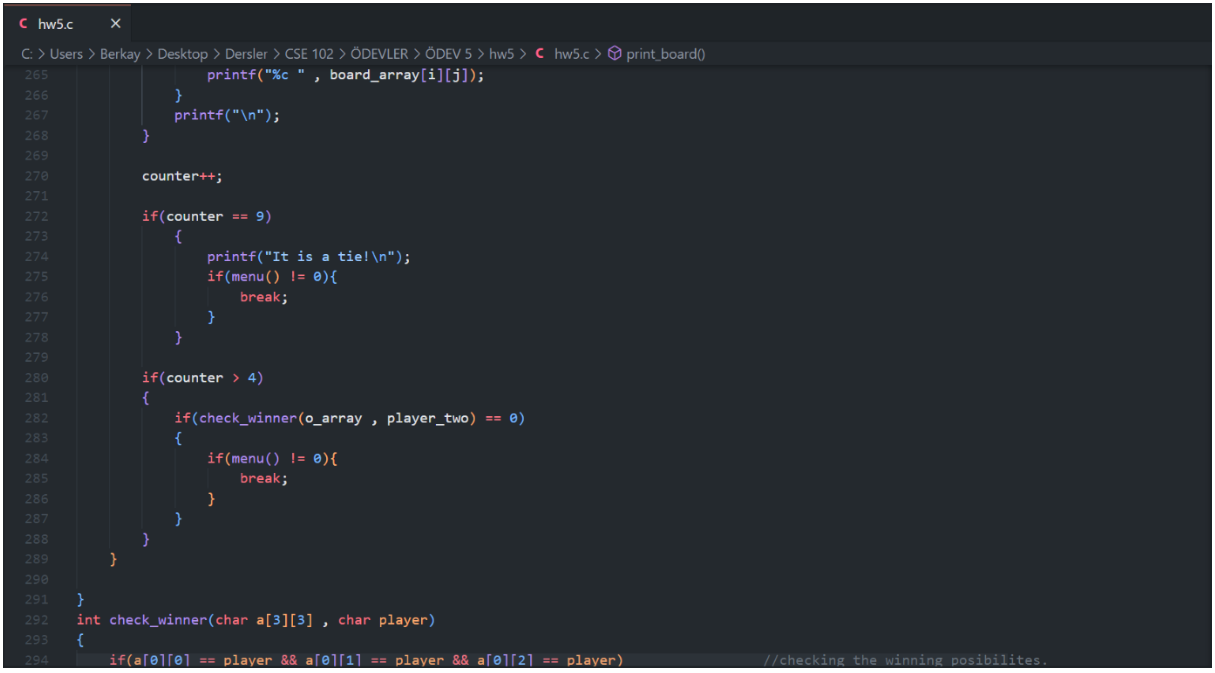

hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

```c
        for(j = 0 ; j < 3 ; j++)
        {
            board_array[i][j] = '_';                                                //filling the arrays.
            x_array[i][j] = '0';
            o_array[i][j] = '0';

        }
    }

    while(1)
    {

        //player 1
        printf("Player 1 (X), enter your move (row , col): ");
            scanf("%d %d" , &x , &y);

            board_array[x][y] = 'X';                                                //filling the coordinate with X in each array.
            x_array[x][y] = 'X';

        for(i = 0 ; i < 3 ; i++)
        {
            for(j = 0 ; j < 3 ; j++)
            {
                printf("%c " , board_array[i][j]);                                  //printing the board array after the selection.
            }
            printf("\n");
        }

        counter++;                                                                  //increasing the counter.

```



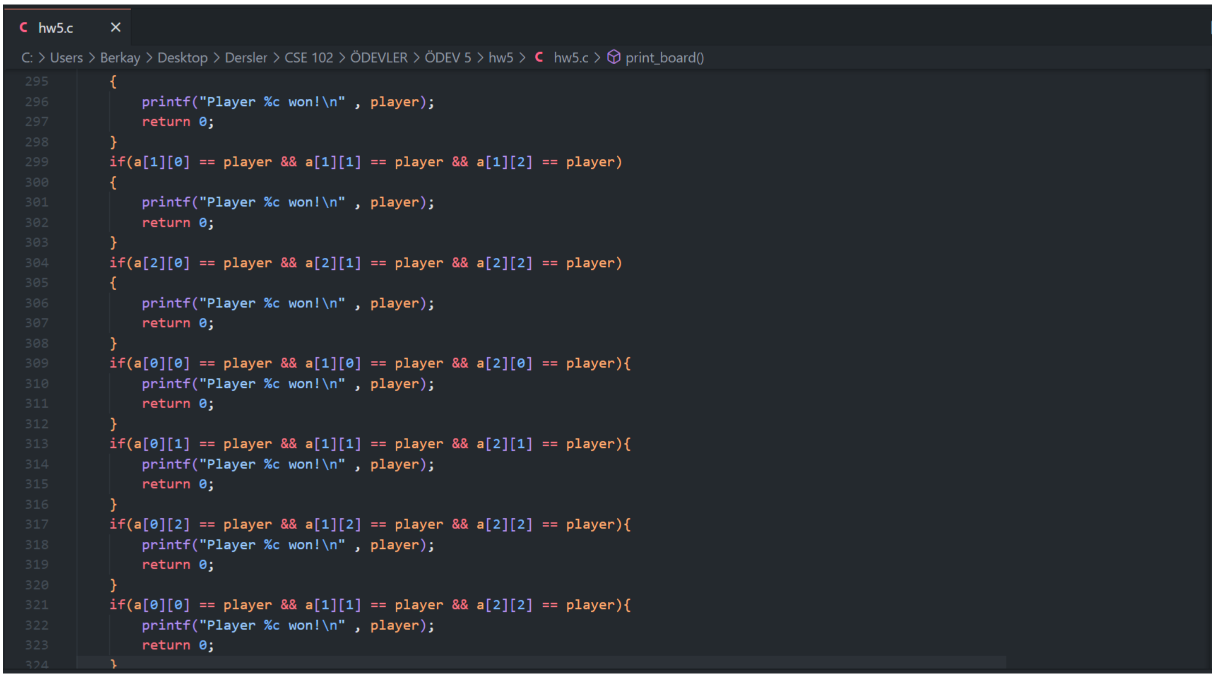
hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

```
        if(counter == 9)                                                             //checking if the game is a tie (play count = 9)
        {
            printf("It is a tie!\n");
            if(menu() != 0){                                                         //sending the user to the menu if it is a tie.
                    break;
                }
        }

        if(counter > 4)                                                              //checking if there is a winner (after the counter
        {                                                                            //is bigger than 4 because filled the arrays with 0)
            if(check_winner(x_array , player_one) == 0)
            {
                if(menu() != 0){                                                     //sending the user to the menu if there is a winner.
                    break;
                }
            }
        }

        //player 2
        printf("Player 2 (O), enter your move (row , col): ");                       //same process with O.
            scanf("%d %d" , &x , &y);

            board_array[x][y] = 'O';
            o_array[x][y] = 'O';

        for(i = 0 ; i < 3 ; i++)
        {
            for(j = 0 ; j < 3 ; j++)
            {
```

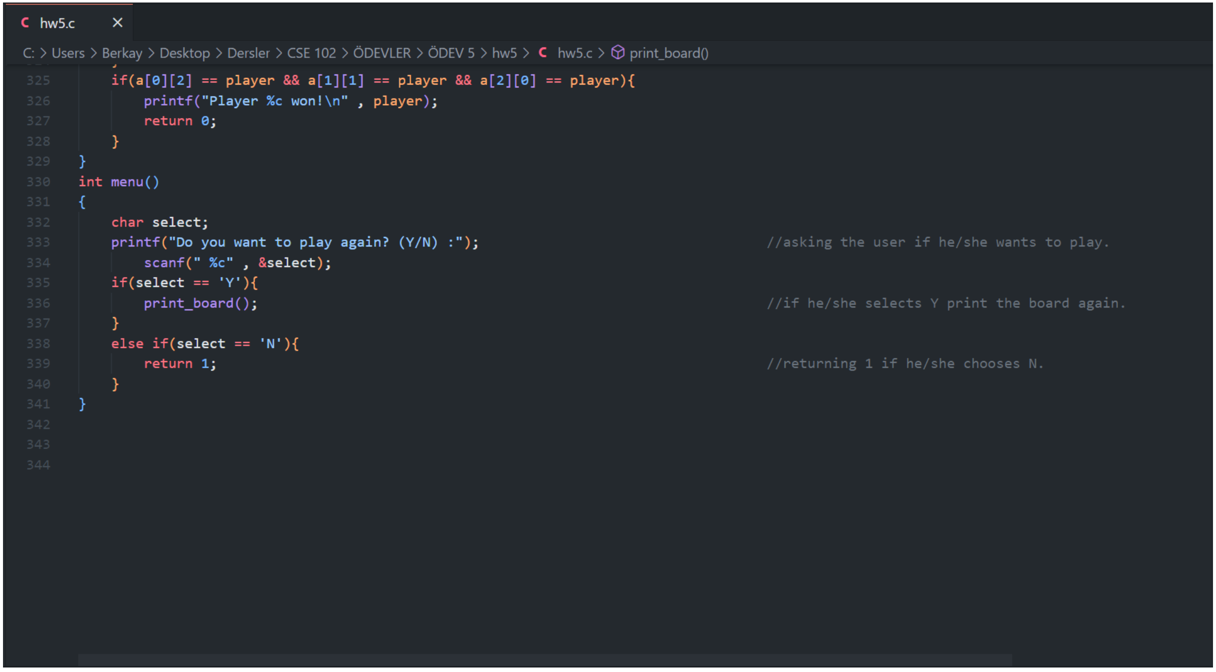


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

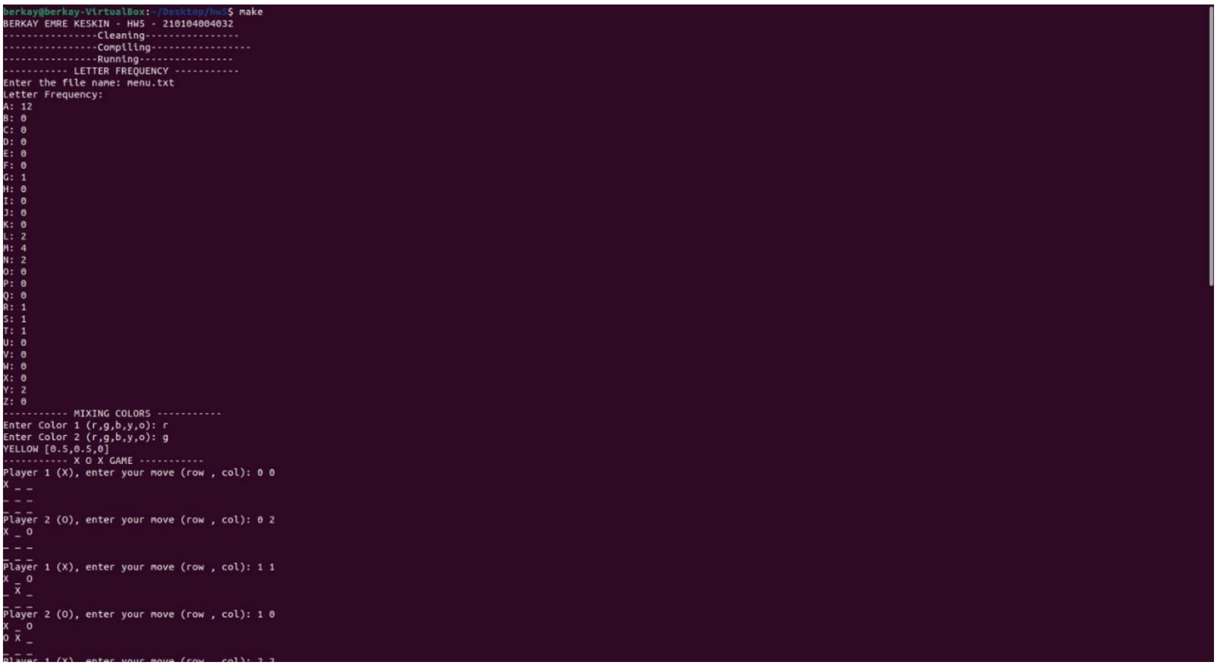
```
                printf("%c " , board_array[i][j]);
            }
            printf("\n");
        }

        counter++;

        if(counter == 9)
            {
                printf("It is a tie!\n");
                if(menu() != 0){
                    break;
                }
            }

        if(counter > 4)
        {
            if(check_winner(o_array , player_two) == 0)
            {
                if(menu() != 0){
                    break;
                }
            }
        }
    }

}
int check_winner(char a[3][3] , char player)
{
    if(a[0][0] == player && a[0][1] == player && a[0][2] == player)                  //checking the winning posibilites.
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

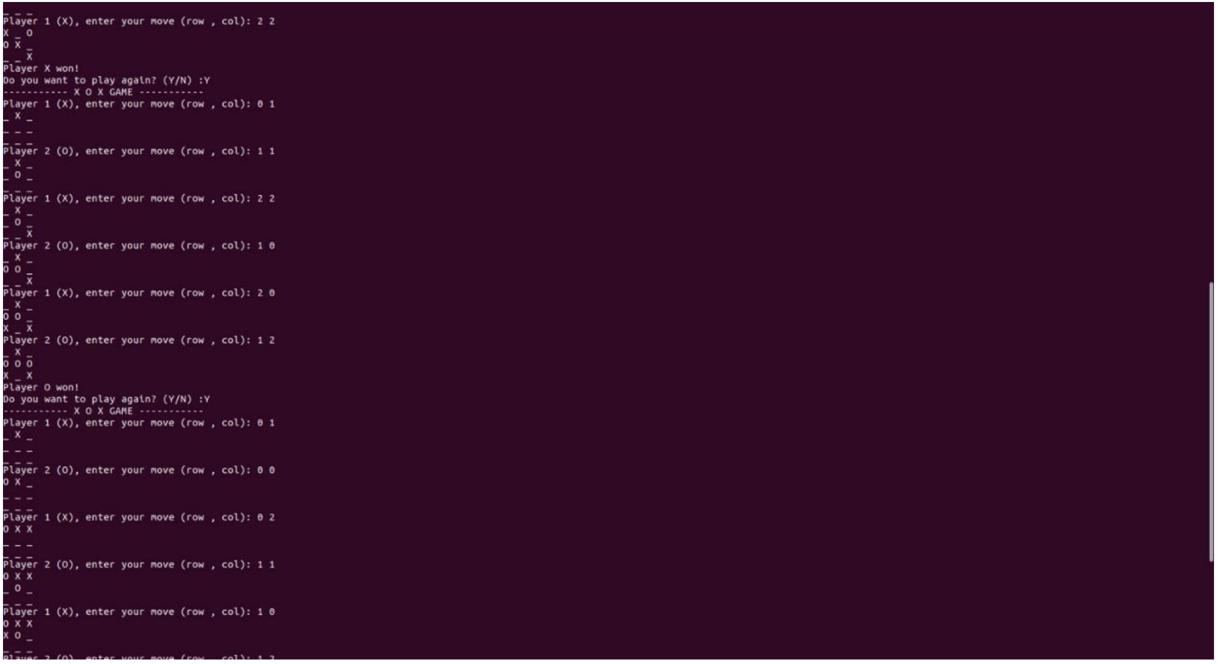
```c
    {
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[1][0] == player && a[1][1] == player && a[1][2] == player)
    {
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[2][0] == player && a[2][1] == player && a[2][2] == player)
    {
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[0][0] == player && a[1][0] == player && a[2][0] == player){
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[0][1] == player && a[1][1] == player && a[2][1] == player){
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[0][2] == player && a[1][2] == player && a[2][2] == player){
        printf("Player %c won!\n" , player);
        return 0;
    }
    if(a[0][0] == player && a[1][1] == player && a[2][2] == player){
        printf("Player %c won!\n" , player);
        return 0;
    }
```


hw5.c
C: > Users > Berkay > Desktop > Dersler > CSE 102 > ÖDEVLER > ÖDEV 5 > hw5 > hw5.c > print_board()

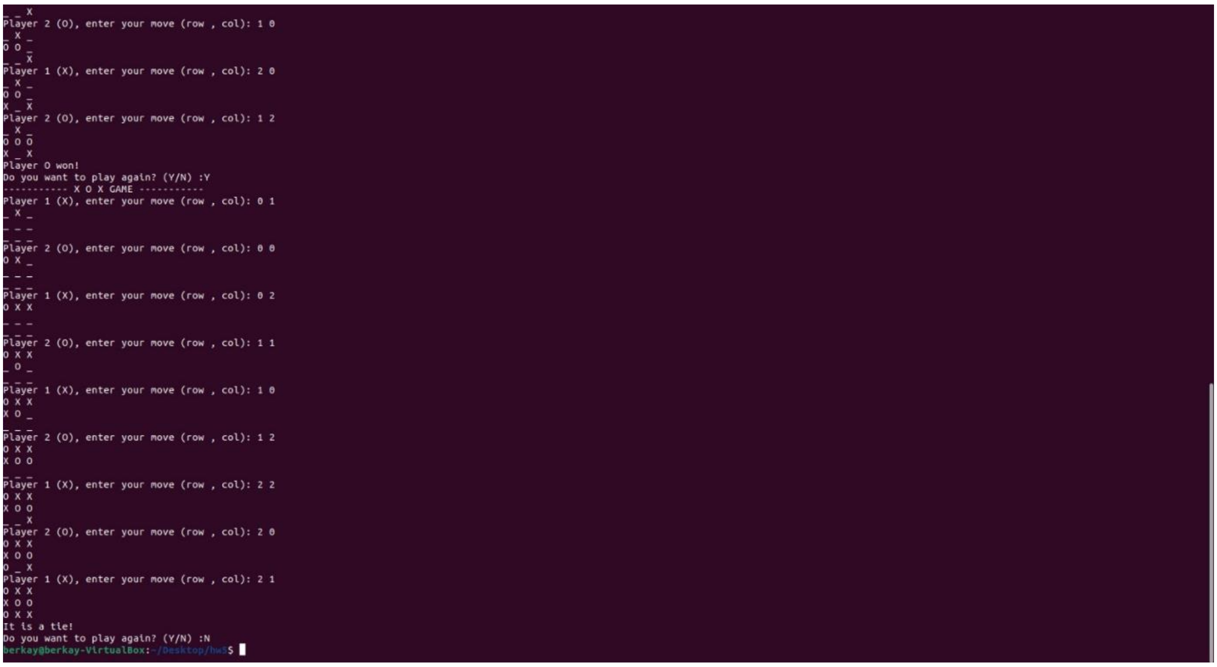
```c
    if(a[0][2] == player && a[1][1] == player && a[2][0] == player){
        printf("Player %c won!\n" , player);
        return 0;
    }
}
int menu()
{
    char select;
    printf("Do you want to play again? (Y/N) :");                              //asking the user if he/she wants to play.
        scanf(" %c" , &select);
    if(select == 'Y'){
        print_board();                                                         //if he/she selects Y print the board again.
    }
    else if(select == 'N'){
        return 1;                                                              //returning 1 if he/she chooses N.
    }
}
```



```
berkay@berkay-VirtualBox:~/Desktop/hw5$ make
BERKAY EMRE KESKIN - HW5 - 210104004032
----------------Cleaning----------------
----------------Compiling-----------------
----------------Running----------------
----------- LETTER FREQUENCY -----------
Enter the file name: menu.txt
Letter Frequency:
A: 12
B: 0
C: 0
D: 0
E: 0
F: 0
G: 1
H: 0
I: 0
J: 0
K: 0
L: 2
M: 4
N: 2
O: 0
P: 0
Q: 0
R: 1
S: 1
T: 1
U: 0
V: 0
W: 0
X: 0
Y: 2
Z: 0
----------- MIXING COLORS -----------
Enter Color 1 (r,g,b,y,o): r
Enter Color 2 (r,g,b,y,o): g
YELLOW [0.5,0.5,0]
----------- X O X GAME -----------
Player 1 (X), enter your move (row , col): 0 0
X _ _
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 0 2
X _ O
_ _ _
_ _ _
Player 1 (X), enter your move (row , col): 1 1
X _ O
_ X _
_ _ _
Player 2 (O), enter your move (row , col): 1 0
X _ O
O X _
_ _ _
Player 1 (X), enter your move (row , col): 2 2
```



```
_ _ _
Player 1 (X), enter your move (row , col): 2 2
X _ O
O X _
_ _ X
Player X won!
Do you want to play again? (Y/N) :Y
----------- X O X GAME -----------
Player 1 (X), enter your move (row , col): 0 1
_ X _
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 1 1
_ X _
_ O _
_ _ _
Player 1 (X), enter your move (row , col): 2 2
_ X _
_ O _
_ _ X
Player 2 (O), enter your move (row , col): 1 0
_ X _
O O _
_ _ X
Player 1 (X), enter your move (row , col): 2 0
_ X _
O O _
X _ X
Player 2 (O), enter your move (row , col): 1 2
_ X _
O O O
X _ X
Player O won!
Do you want to play again? (Y/N) :Y
----------- X O X GAME -----------
Player 1 (X), enter your move (row , col): 0 1
_ X _
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 0 0
O X _
_ _ _
_ _ _
Player 1 (X), enter your move (row , col): 0 2
O X X
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 1 1
O X X
_ O _
_ _ _
Player 1 (X), enter your move (row , col): 1 0
O X X
X O _
_ _ _
Player 2 (O), enter your move (row , col): 1 2
```



```
_ _ X
Player 2 (O), enter your move (row , col): 1 0
_ X _
O O _
_ _ X
Player 1 (X), enter your move (row , col): 2 0
_ X _
O O _
X _ X
Player 2 (O), enter your move (row , col): 1 2
_ X _
O O O
X _ X
Player O won!
Do you want to play again? (Y/N) :Y
----------- X O X GAME -----------
Player 1 (X), enter your move (row , col): 0 1
_ X _
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 0 0
O X _
_ _ _
_ _ _
Player 1 (X), enter your move (row , col): 0 2
O X X
_ _ _
_ _ _
Player 2 (O), enter your move (row , col): 1 1
O X X
_ O _
_ _ _
Player 1 (X), enter your move (row , col): 1 0
O X X
X O _
_ _ _
Player 2 (O), enter your move (row , col): 1 2
O X X
X O O
_ _ _
Player 1 (X), enter your move (row , col): 2 2
O X X
X O O
_ _ X
Player 2 (O), enter your move (row , col): 2 0
O X X
X O O
O _ X
Player 1 (X), enter your move (row , col): 2 1
O X X
X O O
O X X
It is a tie!
Do you want to play again? (Y/N) :N
berkay@berkay-VirtualBox:~/Desktop/hw5$
```