# ASSINGMENT 1

Progress #1

#include <stdio.h>

#include <string.h>

int main() {

char player1[25], player2[25];//player name can't be longer than 25 characters and can't have space character

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player1);

printf("Enter name of player 2:");

scanf("%s", player2);

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player1, player2);

return 0;

}



**Modified**

#include <stdio.h>

#include <string.h>

int main() {

char player1[25], player2[25];//player name can't be longer than 25 characters and can't have space character

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player1);

printf("Enter name of player 2:");

scanf("%s", player2);

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player1, player2);

printf("\n");

printf("\n# Objective: Capture Opponent's token");

printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");

printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");

printf("\n# Points: Number of tokens cut by the player is the player's score.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

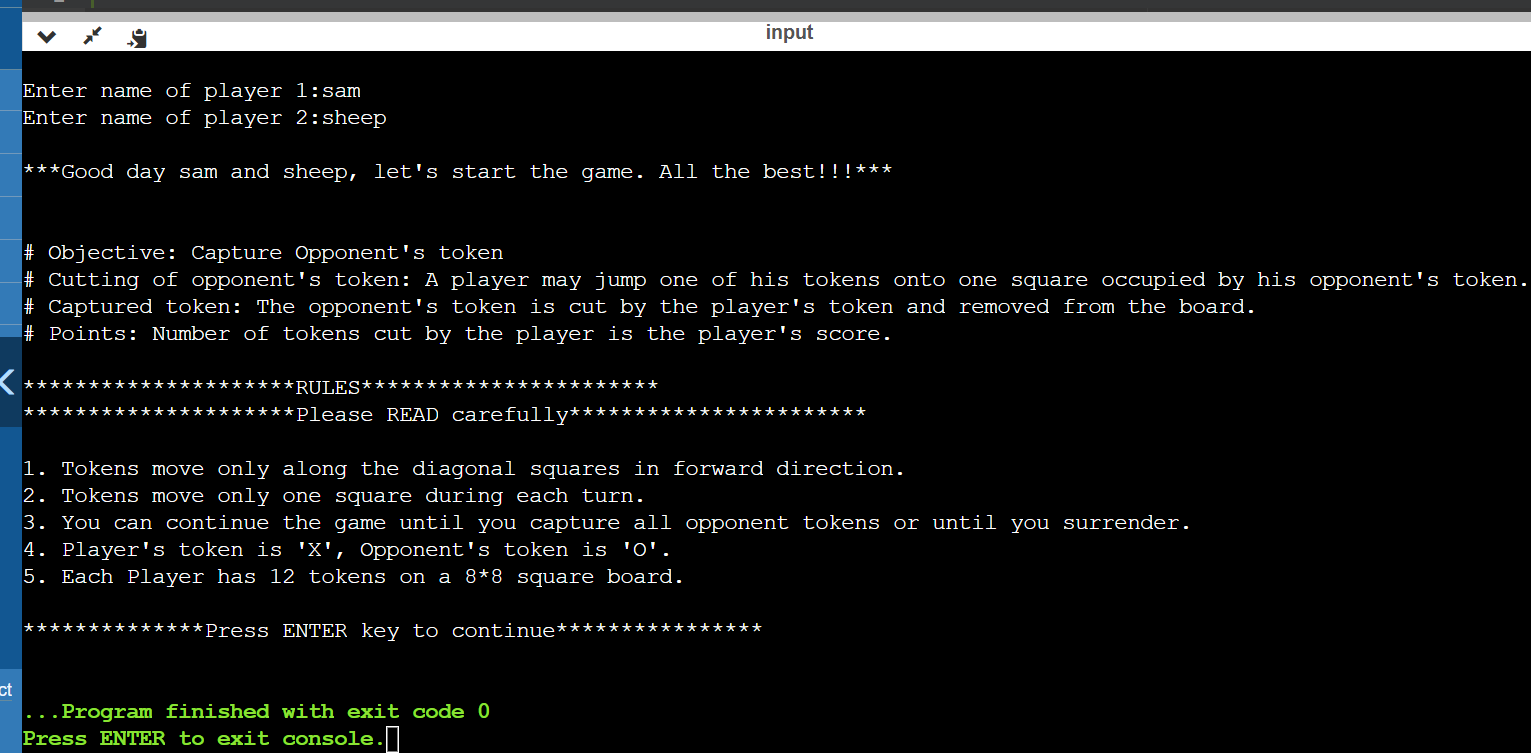
printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n4. Player's token is 'X', Opponent's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

return 0;

}



Progress #2

#include<stdio.h>

int main()

{

char a[8][8]= { {'X',' ','X',' ','X',' ','X',' '},

{' ','X',' ','X',' ','X',' ','X'},

{'X',' ','X',' ','X',' ','X',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','O',' ','O',' ','O',' ','O'},

{'O',' ','O',' ','O',' ','O',' '},

{' ','O',' ','O',' ','O',' ','O'} };

int i,j;

printf(" --- --- --- --- --- --- --- ---");

for(i=0;i<8;i++)

{

printf("\n");

for(j=0;j<8;j++)

{

printf("| %c ",a[i][j]);

}

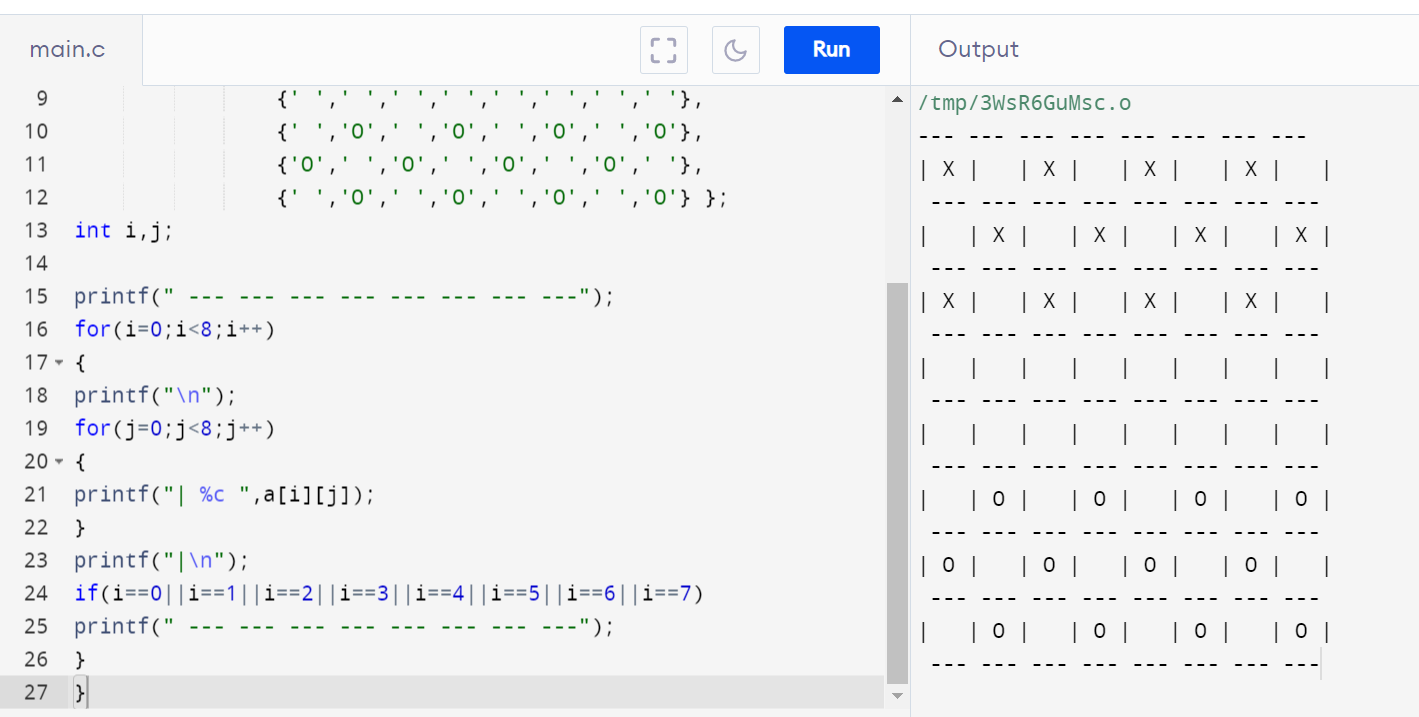
printf("|\n");

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

printf(" --- --- --- --- --- --- --- ---");

}

}



**Modified**

#include<stdio.h>

int main()

{

char a[8][8]={{'O',' ','O',' ','O',' ','O',' '},{' ','O',' ','O',' ','O',' ','O'},{'O',' ','O',' ','O',' ','O',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','X',' ','X',' ','X',' ','X'},{'X',' ','X',' ','X',' ','X',' '},{' ','X',' ','X',' ','X',' ','X'}};

int i,j;

printf(" --- --- --- --- --- --- --- --- ");

for(i=0;i<8;i++)

{

printf(" \n");

for(j=0;j<8;j++)

{

printf("| %c ",a[i][j]);

}

printf("| %d \n",-(i-7));

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

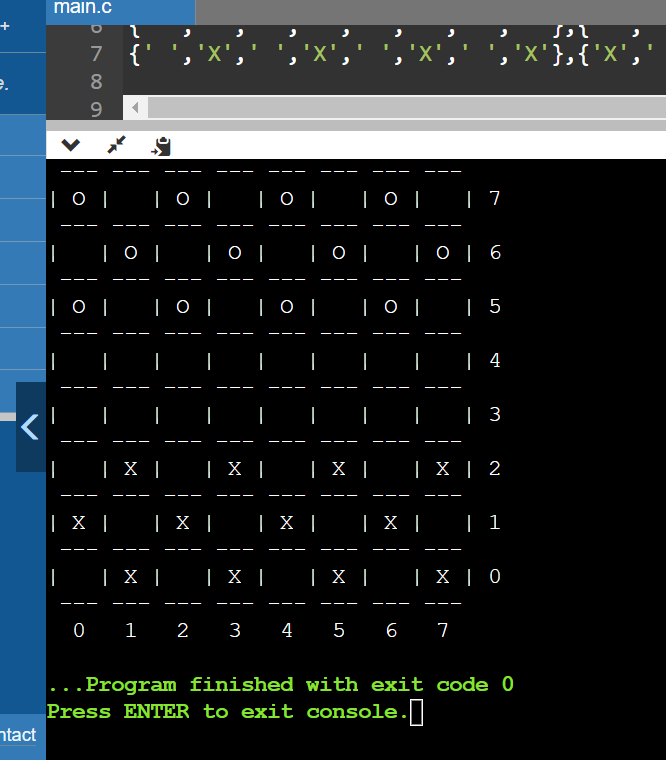
printf(" --- --- --- --- --- --- --- --- ");

}

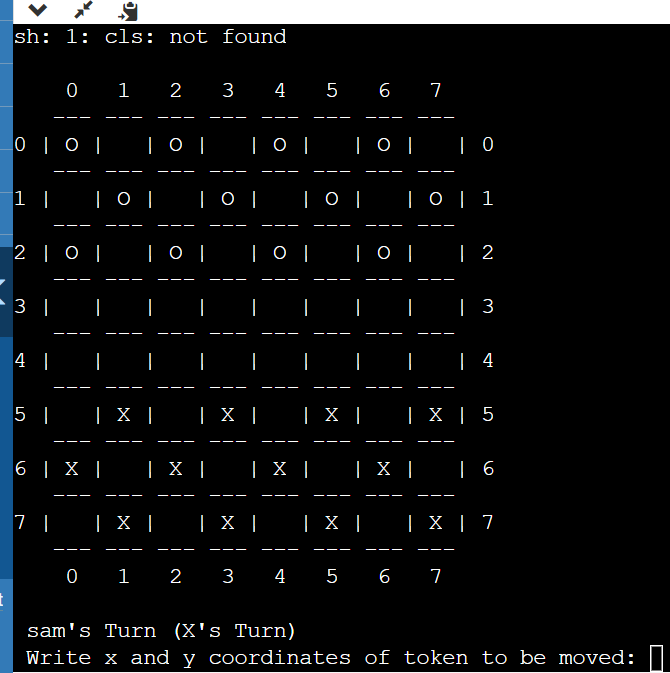
printf("\n 0 1 2 3 4 5 6 7");

return 0;

}



**Latest correct board**

****

**Full #1**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//coding for checker board and tokens using user-defined function

void draw\_board(void)

{

char a[8][8]={{'X',' ','X',' ','X',' ','X',' '},{' ','X',' ','X',' ','X',' ','X'},{'X',' ','X',' ','X',' ','X',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','O',' ','O',' ','O',' ','O'},{'O',' ','O',' ','O',' ','O',' '},{' ','O',' ','O',' ','O',' ','O'}};

int i,j;

printf(" --- --- --- --- --- --- --- --- ");

for(i=0;i<8;i++)

{

printf(" \n");

for(j=0;j<8;j++)

{

printf("| %c ",a[i][j]);

}

printf("| %d \n",-(i-7));

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

printf(" --- --- --- --- --- --- --- --- ");

}

printf("\n 0 1 2 3 4 5 6 7");

}

int main() {

char player1[25], player2[25];//player name can't be longer than 25 characters and can't have space character

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player1);

printf("Enter name of player 2:");

scanf("%s", player2);

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player1, player2);

printf("\n");

draw\_board();

return 0;

}

Progress #3

Clear screen example (works in codeblocks/TurboC compilers only)-

#include <stdio.h>

#include <stdlib.h>

int main(){

printf("Press any key to clear the screen\n");

printf("A B C D E F G H\n");

printf("1 \n 2 \n 3 \n 4 \n");

getchar();

system("cls"); // For Windows use system("cls");

printf("1 \n 2 \n 3 \n 4 \n");

printf("A B C D E F G H\n");

getchar();

return 0;

}

For GCC/G++ compilers in Linux

#include <stdio.h>

#include <stdlib.h> /\*for system() function\*/

void screen\_clear(void){

//printf("\nPlease press ENTER key\n");

//getchar();

system("clear");

}

int main()

{

int num=100;

printf("Hello");

getchar();

/\*Use after declaration section\*/

//system("clear"); /\*clear output screen\*/

screen\_clear();

printf("value of num: %d\n",num);

getchar();

return 0;

}

**Full #2**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void screen\_clear(void){

//printf("\nPlease press ENTER key\n");

//getchar();

system("clear");

}

//coding for checker board and tokens using user-defined function

void draw\_board(void)

{

char checkers[8][8]={{'X',' ','X',' ','X',' ','X',' '},{' ','X',' ','X',' ','X',' ','X'},{'X',' ','X',' ','X',' ','X',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','O',' ','O',' ','O',' ','O'},{'O',' ','O',' ','O',' ','O',' '},{' ','O',' ','O',' ','O',' ','O'}};

int i,j;

printf(" --- --- --- --- --- --- --- --- ");

for(i=0;i<8;i++)

{

printf(" \n");

for(j=0;j<8;j++)

{

printf("| %c ",checkers[i][j]);

}

printf("| %d \n",-(i-7));

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

printf(" --- --- --- --- --- --- --- --- ");

}

printf("\n 0 1 2 3 4 5 6 7");

}

int main() {

char player1[25], player2[25];//player name can't be longer than 25 characters and can't have space character

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player1);

printf("Enter name of player 2:");

scanf("%s", player2);

getchar();

screen\_clear();

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player1, player2);

printf("\n");

printf("\n# Objective: Capture Opponent's token");

printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");

printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");

printf("\n# Points: Number of tokens cut by the player is the player's score.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n4. Player's token is 'X', Opponent's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

draw\_board();

return 0;

}

Progress #4

Displaying player scores (Invalid Version)

#include <stdio.h>

#include <stdlib.h>

struct players

{

int SCORE;

char NAME[25];

}player[100],k;

void main()

{

int i,j,n;

printf("Enter no.of.players: ");

scanf("%d", &n);

printf("Enter player's NAME , SCORE\n");

for(i=0;i<n;i++)

{

scanf("%s %d", player[i].NAME,&player[i].SCORE);

}

for(i=0;i<n;i++)

{

for(j=0;j<n-1;j++)

{

if(player[j].SCORE<player[j+1].SCORE)

{

k=player[j];

player[j]=player[j+1];

player[j+1]=k;

}

}

}

printf("\nPlayers Score\n");

printf("\nRANK\t\t\tNAME\t\t\tSCORE\n");

printf("------------------------------------------------------------\n");

printf("TOP 3 PLAYERS \n");

for(i=0;i<n;i++)

{

if(i==3)

{

printf("---------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------\n");

printf("%d\t\t\t%s\t\t\t%d\n",i+1,player[i].NAME,player[i].SCORE);

}

else

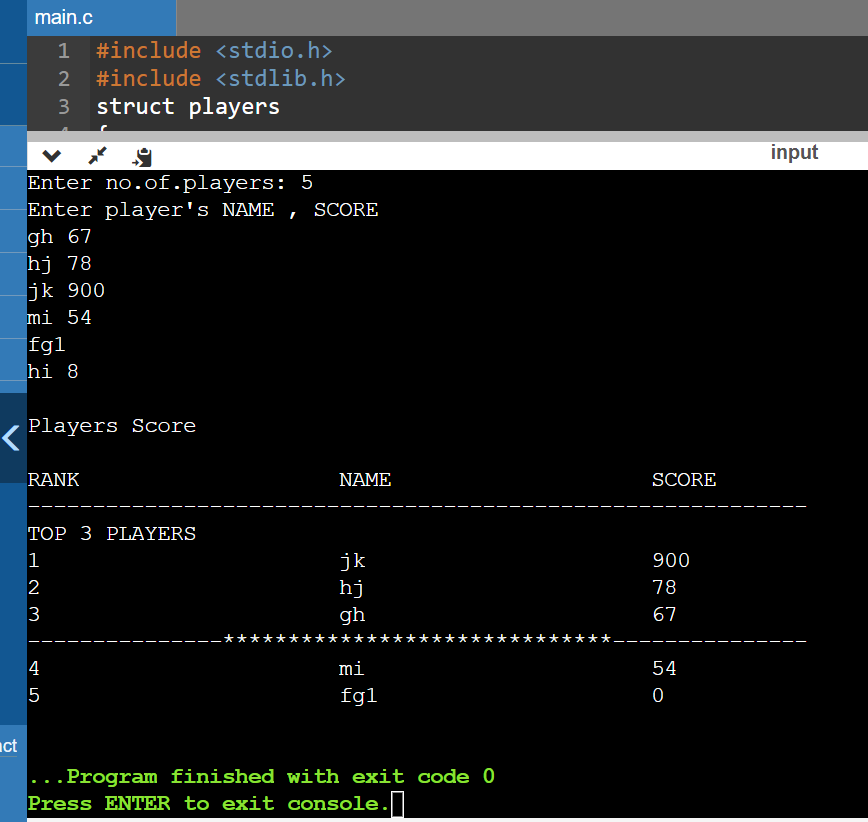
{

printf("%d\t\t\t%s\t\t\t%d\n",i+1,player[i].NAME,player[i].SCORE);

}

}

}



Progress #5

#include<stdio.h>  
#include <stdlib.h>  
#include <string.h>  
struct token{  
int token\_p;    //for player  
int token\_opp;  //for opponent  
};  
     char checkers[8][8]={  
        {'X',' ','X',' ','X',' ','X',' '},  
        {' ','X',' ','X',' ','X',' ','X'},  
        {'X',' ','X',' ','X',' ','X',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ','O',' ','O',' ','O',' ','O'},  
        {'O',' ','O',' ','O',' ','O',' '},  
        {' ','O',' ','O',' ','O',' ','O'}  
        };  
//function to clear screen  
void screen\_clear(void){  
     system("clear");  
 }  
  
 void move\_O()//for opponent  
{  
    struct token old\_x, old\_y, new\_x, new\_y;  
  
  
    printf("Write x and y coordinates of token to be moved: ");  
    scanf("%d %d" ,&old\_x.token\_opp ,&old\_y.token\_opp);  
  
    printf("Write x and y coordinates of square where token is to be placed: ");  
    scanf("%d %d" ,&new\_x.token\_opp ,&new\_y.token\_opp);  
}  
  
void move\_X(int x, int y,int new\_x, int new\_y)//for player  
{  
  
   int i,j;  
    printf(" --- --- --- --- --- --- --- --- ");  
    for(i=0;i<8;i++)  
    {  
        printf(" \n");  
        for(j=0;j<8;j++)  
        {  
            printf("| %c ",checkers[i][j]);  
        }  
        printf("| %d \n",-(i-7));  
        if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf(" --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n  0   1   2   3   4   5   6   7");  
  
  
}  
  
void get\_turn(int Turn, char \*player1, char \*player2){  
  
    struct token old\_x, old\_y, new\_x, new\_y;  
    for(Turn=1; ;Turn++)  
    {  
        if(Turn%2==0)  
        {  
            printf("\n %s's Turn (O's Turn)",player2);  
  
            move\_O();  
        }  
        else{  
            printf("\n %s's Turn (X's Turn)",player1);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old\_x.token\_p ,&old\_y.token\_p);  
  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new\_x.token\_p ,&new\_y.token\_p);  
            move\_X(old\_x.token\_p,old\_y.token\_p,new\_x.token\_p,new\_y.token\_p);  
        }  
  
    }  
}  
  
//9 9 is surrender code  
void end\_game(struct token old\_x, struct token old\_y){  
    if(old\_x.token\_opp==9 && old\_y.token\_opp==9){  
        screen\_clear();  
  
      printf("X won!");  
    }  
  
    else if(old\_x.token\_p==9 && old\_y.token\_p==9){  
        screen\_clear();  
  
        printf("O won!");  
    }  
}  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//coding for checker board and tokens using user-defined function  
void draw\_board(void)  
{  
  
    int i,j;  
    printf(" --- --- --- --- --- --- --- --- ");  
    for(i=0;i<8;i++)  
    {  
        printf(" \n");  
        for(j=0;j<8;j++)  
        {  
            printf("| %c ",checkers[i][j]);  
        }  
        printf("| %d \n",-(i-7));  
        if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf(" --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n  0   1   2   3   4   5   6   7");  
}  
  
int main() {  
    char player1[25], player2[25];//player name can't be longer than 25 characters and can't have space character  
    int turn=1;  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("\nEnter name of player 1:");  
    scanf("%s", player1);  
  
    printf("Enter name of player 2:");  
    scanf("%s", player2);  
  
    getchar();  
    screen\_clear();  
  
    printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player1, player2);  
    printf("\n");  
  
    printf("\n# Objective: Capture Opponent's token");  
    printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");  
    printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");  
    printf("\n# Points: Number of tokens cut by the player is the player's score.\n");  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player's token is 'X', Opponent's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
    draw\_board();  
    get\_turn(turn, player1, player2);  
    return 0;  
}

Progress #6

#include<stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
//Global variables  
char current\_Player[25];//used to see which player won  
int end\_flag=0;//used in end\_game() function  
  
struct token{  
int x, y;// x and y coordinate  
};  
  
struct player{  
int count, current\_flag;//count for score, and for whoever current\_flag=1, that is current player  
char name[25];//name of player  
};  
  
     char checkers[8][8]={  
        {'O',' ','O',' ','O',' ','O',' '},  
        {' ','O',' ','O',' ','O',' ','O'},  
        {'O',' ','O',' ','O',' ','O',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ','X',' ','X',' ','X',' ','X'},  
        {'X',' ','X',' ','X',' ','X',' '},  
        {' ','X',' ','X',' ','X',' ','X'}  
        };  
  
//function to clear screen  
void screen\_clear(void){  
     system("cls");  
 }  
  
 //9 9 is surrender code  
int end\_game(struct token old){  
    if(old.x==9 && old.y==9){  
        end\_flag=1;  
        screen\_clear();  
  
      printf("\nGAME OVER!\n");  
      printf("\n%s WON!\n",current\_Player);  
    }  
    return end\_flag;  
}  
  
 /\*void move\_O(int old\_x, int old\_y,int new\_x, int new\_y)//for opponent  
{  
    struct token old\_x, old\_y, new\_x, new\_y;  
  
  
    printf("Write x and y coordinates of token to be moved: ");  
    scanf("%d %d" ,&old\_x.token\_opp ,&old\_y.token\_opp);  
  
    printf("Write x and y coordinates of square where token is to be placed: ");  
    scanf("%d %d" ,&new\_x.token\_opp ,&new\_y.token\_opp);  
}\*/  
  
void move\_X(struct token old, struct token new)//for player  
{  
  
   int i,j;  
  
   if(end\_game(old)==0)  
{  
        printf(" --- --- --- --- --- --- --- --- ");  
        for(i=0;i<8;i++)  
        {  
                printf(" \n");  
                for(j=0;j<8;j++)  
                {  
                    printf("| %c ",checkers[i][j]);  
                }  
                printf("| %d \n",-(i-7));  
                if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
                    printf(" --- --- --- --- --- --- --- --- ");  
            }  
            printf("\n  0   1   2   3   4   5   6   7");  
   }  
  
}  
  
int get\_turn(int Turn, struct player player\_X, struct player player\_O){  
  
    struct token old, new;  
    int rem =0;  
    for(Turn=1; ;Turn++)  
    {  
  
        if(Turn%2==0)  
        {  
            printf("\n %s's Turn (O's Turn)",player\_O.name);  
  
            //move\_O();  
        }  
        else{  
            printf("\n %s's Turn (X's Turn)",player\_X.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
            move\_X(old, new);  
            rem = 1;  
        }  
  
    }  
    return rem;  
}  
  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//coding for checker board and tokens using user-defined function  
void draw\_board(void)  
{  
  
    int i,j;  
    printf(" --- --- --- --- --- --- --- --- ");  
    for(i=0;i<8;i++)  
    {  
        printf(" \n");  
        for(j=0;j<8;j++)  
        {  
            printf("| %c ",checkers[i][j]);  
        }  
        printf("| %d \n",-(i-7));  
        if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf(" --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n  0   1   2   3   4   5   6   7");  
}  
  
int main() {  
    //char player1[25], player2[25];  
    struct player player\_X, player\_O;  
    int turn=1;  
  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("\nEnter name of player 1:");  
    scanf("%s", player\_X.name);  
  
    printf("Enter name of player 2:");  
    scanf("%s", player\_O.name);  
  
    getchar();  
    screen\_clear();  
  
    printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);  
    printf("\n");  
  
    printf("\n# Objective: Capture Opponent's token");  
    printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");  
    printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");  
    printf("\n# Points: Number of tokens cut by the player is the player's score.\n");  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
  
    draw\_board();  
  
  
    /\* if (get\_turn(turn, player\_X, player\_O) == 0){  
        strcpy(current\_Player,player\_X);  
    }  
    else{  
       strcpy(current\_Player,player\_O);  
    }  
       //puts(current\_Player);  
       printf("%s",current\_Player);\*/  
  
    return 0;  
}

Progress #7

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Welcome to draught programed by Samuela, Sakthi, Merlin, and Deepiga. \*  
 \*                                                                       \*  
 \*                           AI & DS Department                          \*  
 \*                    Coimbatore Institute of Technology                 \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
//Global variables  
char current\_Player[25];  
//char winner\_Player[25];//used to see which player won  
int end\_flag=0;//used in end\_game() function  
  
struct token{  
int x, y;// x and y coordinate  
};  
  
struct player{  
int count;//count for score  
int current\_flag;//for whoever current\_flag=1, that is current player  
char name[25];//name of player  
};  
  
    struct player player\_X, player\_O;  
  
     char checkers[8][8]={  
        {'O',' ','O',' ','O',' ','O',' '},  
        {' ','O',' ','O',' ','O',' ','O'},  
        {'O',' ','O',' ','O',' ','O',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ',' ',' ',' ',' ',' ',' ',' '},  
        {' ','X',' ','X',' ','X',' ','X'},  
        {'X',' ','X',' ','X',' ','X',' '},  
        {' ','X',' ','X',' ','X',' ','X'}  
        };  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of declaration of Global variables \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//function to clear screen  
void screen\_clear(void){  
     system("cls");  
 }  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of screen\_clear() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//9 9 is surrender code  
int end\_game(struct token old){  
  
    if(old.x==9 && old.y==9)  
    {  
        end\_flag=1;  
        screen\_clear();  
        printf("\nGAME OVER!\n");  
        if(strcmp(current\_Player,player\_X.name)==0)  
        {  
            printf("\n%s WON!\n",player\_O.name);  
        }  
        else  
        {  
            printf("\n%s WON!\n",player\_X.name);  
        }  
    }  
    return end\_flag;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of end\_game() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
 void move\_O(struct token old, struct token new)//for player2  
{  
    int i,j;  
  
   checkers[new.x][new.y]=checkers[old.x][old.y];  
   checkers[old.x][old.y]=' ';  
    strcpy(current\_Player,player\_O.name);  
   if(end\_game(old)==0)  
    {  
        printf(" --- --- --- --- --- --- --- --- ");  
        for(i=0;i<8;i++)  
        {  
                printf(" \n");  
                for(j=0;j<8;j++)  
                {  
                    printf("| %c ",checkers[i][j]);  
                }  
                printf("| %d \n",-(i-7));  
                if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
                    printf(" --- --- --- --- --- --- --- --- ");  
            }  
            printf("\n  0   1   2   3   4   5   6   7");  
   }  
  
   if(checkers[new.x][new.y]=='X')  
    player\_O.count++;  
  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_O() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
void move\_X(struct token old, struct token new)//for player1  
{  
    int i,j;  
    checkers[new.x][new.y]=checkers[old.x][old.y];  
    checkers[old.x][old.y]=' ';  
    strcpy(current\_Player,player\_X.name);  
    if(end\_game(old)==0)  
    {  
        printf(" --- --- --- --- --- --- --- --- ");  
        for(i=0;i<8;i++)  
        {  
            printf(" \n");  
            for(j=0;j<8;j++)  
            {  
                printf("| %c ",checkers[i][j]);  
            }  
            printf("| %d \n",-(i-7));  
            if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf(" --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n  0   1   2   3   4   5   6   7");  
    }  
    if(checkers[new.x][new.y]=='O')  
    player\_X.count++;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_X() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int get\_turn(int Turn, struct player player\_X, struct player player\_O){  
  
    struct token old, new;  
    int rem =0;  
    for(Turn=1; ;Turn++)  
    {  
        if(end\_flag==1)  
            break;  
  
        if(Turn%2==0)  
        {  
            printf("\n %s's Turn (O's Turn)",player\_O.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
            move\_O(old, new);  
        }  
        else  
        {  
            printf("\n %s's Turn (X's Turn)",player\_X.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
            move\_X(old, new);  
            rem = 1;  
        }  
    }  
    return rem;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of get\_turn() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//coding for checker board and tokens using user-defined function  
void draw\_board(void)  
{  
  
    int i,j;  
    printf(" --- --- --- --- --- --- --- --- ");  
    for(i=0;i<8;i++)  
    {  
        printf(" \n");  
        for(j=0;j<8;j++)  
        {  
            printf("| %c ",checkers[i][j]);  
        }  
        printf("| %d \n",-(i-7));  
        if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf(" --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n  0   1   2   3   4   5   6   7");  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of draw\_board() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Start of main() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int main() {  
    //char player1[25], player2[25];  
  
    int turn=2;  
  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("\nEnter name of player 1:");  
    scanf("%s", player\_X.name);  
  
    printf("Enter name of player 2:");  
    scanf("%s", player\_O.name);  
  
    getchar();  
    screen\_clear();  
  
    printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);  
    printf("\n");  
  
    printf("\n# Objective: Capture Opponent's token");  
    printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");  
    printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");  
    printf("\n# Points: Number of tokens cut by the player is the player's score.\n");  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
  
    draw\_board();  
    turn = get\_turn(turn, player\_X, player\_O);  
  
  
    return 0;  
}

Progress #8

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Welcome to draught programed by Samuela, Sakthi, Merlin, and Deepiga. \*  
 \*                                                                       \*  
 \*                           AI & DS Department                          \*  
 \*                    Coimbatore Institute of Technology                 \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
//Global variables  
char current\_Player[25];  
//char winner\_Player[25];//used to see which player won  
int end\_flag=0;//used in end\_game() function  
  
struct token{  
int x, y;// x and y coordinate  
int king\_flag; //for whoever king\_flag=1, that token is king  
};  
//struct token token\_X, token\_O;  
  
struct player{  
int count;//count for score  
int current\_flag;//for whoever current\_flag=1, that is current player  
char name[25];//name of player  
};  
  
struct player player\_X, player\_O;  
  
 char checkers[8][8]={  
    {'O',' ','O',' ','O',' ','O',' '},  
    {' ','O',' ','O',' ','O',' ','O'},  
    {'O',' ','O',' ','O',' ','O',' '},  
    {' ',' ',' ',' ',' ',' ',' ',' '},  
    {' ',' ',' ',' ',' ',' ',' ',' '},  
    {' ','X',' ','X',' ','X',' ','X'},  
    {'X',' ','X',' ','X',' ','X',' '},  
    {' ','X',' ','X',' ','X',' ','X'}  
    };  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of declaration of Global variables \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//function to clear screen  
void screen\_clear(void){  
     system("cls");  
 }  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of screen\_clear() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//9 9 is surrender code  
int end\_game(struct token old){  
  
    if(old.x==9 && old.y==9)  
    {  
        end\_flag=1;  
        screen\_clear();  
        printf("\nGAME OVER!\n");  
        if(strcmp(current\_Player,player\_X.name)==0)  
        {  
            printf("\n%s WON!\n",player\_O.name);  
        }  
        else  
        {  
            printf("\n%s WON!\n",player\_X.name);  
        }  
    }  
    return end\_flag;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of end\_game() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//coding for checker board and tokens using user-defined function  
void draw\_board(void)  
{  
  
    int i,j;  
        printf("\n    0   1   2   3   4   5   6   7 \n");  
        printf("   --- --- --- --- --- --- --- --- ");  
        for(i=0;i<8;i++)  
        {  
            printf(" \n%d ",i);  
            for(j=0;j<8;j++)  
            {  
                printf("| %c ",checkers[i][j]);  
            }  
            printf("| %d \n",i);  
            if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf("   --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n    0   1   2   3   4   5   6   7 \n");  
}  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of draw\_board() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
 void move\_O(struct token old, struct token new)//for player2  
{  
   checkers[new.x][new.y]=checkers[old.x][old.y];  
   checkers[old.x][old.y]=' ';  
  
   printf("\ncheckers[%d][%d]=%c \n",new.x,new.y,checkers[new.x][new.y]);  
   printf("checkers[%d][%d]=%c \n",old.x,old.y, checkers[old.x][old.y]);  
  
    strcpy(current\_Player,player\_O.name);  
   if(end\_game(old)==0)  
    {  
        draw\_board();  
   }  
  
   if(checkers[new.x][new.y]=='X')  
    player\_O.count++;  
  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_O() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
void move\_X(struct token old, struct token new)//for player1  
{  
    checkers[new.x][new.y]=checkers[old.x][old.y];  
    checkers[old.x][old.y]=' ';  
  
   printf("\ncheckers[%d][%d]=%c \n",new.x,new.y,checkers[new.x][new.y]);  
   printf("checkers[%d][%d]=%c \n",old.x,old.y, checkers[old.x][old.y]);  
  
    strcpy(current\_Player,player\_X.name);  
    if(end\_game(old)==0)  
    {  
       draw\_board();  
    }  
    if(checkers[new.x][new.y]=='O')  
    player\_X.count++;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_X() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int get\_turn(int Turn, struct player player\_X, struct player player\_O){  
  
    struct token old, new;  
    int rem =0;  
    for(Turn=1; ;Turn++)  
    {  
        if(end\_flag==1)  
            break;  
  
        if(Turn%2==0)  
        {  
            printf("\n %s's Turn (O's Turn)",player\_O.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
            move\_O(old, new);  
        }  
        else  
        {  
            printf("\n %s's Turn (X's Turn)",player\_X.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
            move\_X(old, new);  
            rem = 1;  
        }  
    }  
    return rem;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of get\_turn() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Start of main() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int main() {  
    //char player1[25], player2[25];  
  
    int turn=2;  
  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("\nEnter name of player 1:");  
    scanf("%s", player\_X.name);  
  
    printf("Enter name of player 2:");  
    scanf("%s", player\_O.name);  
  
    getchar();  
    screen\_clear();  
  
    printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);  
    printf("\n");  
  
    printf("\n# Objective: Capture Opponent's token");  
    printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");  
    printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");  
    printf("\n# Points: Number of tokens cut by the player is the player's score.\n");  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
  
    draw\_board();  
    turn = get\_turn(turn, player\_X, player\_O);  
  
  
    return 0;  
}

Progress #9

Colored text (for GCC/G++)

#include <stdio.h>

void red(){

printf("\033[1;31m");

}

void reset();

int main()

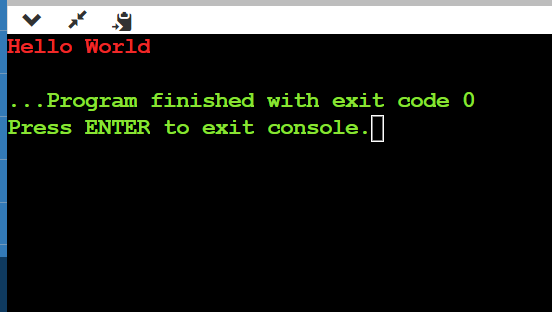
{

red();

printf("Hello World");

return 0;

}

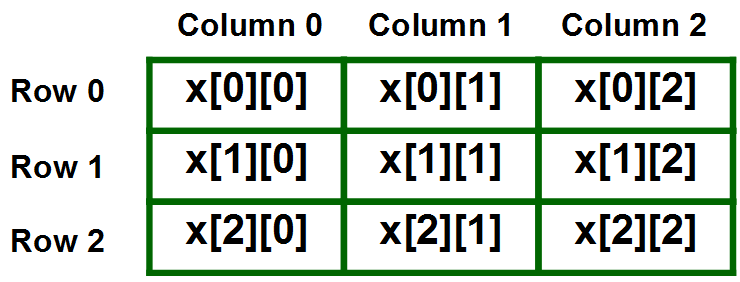


NOTE:- Reason why x and y are reverse in program (because of way in array, look below)

Rows are actually Y-axis, and columns are X-axis

So x[0][0], x[0][1], x[0][2] are actually reverse.

Like x[0][1] is x[row 0 (y coordinate or coordinate on y-axis)][column 1(x coordinate or coordinate on x-axis)]



Progress #10

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Welcome to draught programed by Samuela, Sakthi, Merlin, and Deepiga. \*  
 \*                                                                       \*  
 \*                           AI & DS Department                          \*  
 \*                    Coimbatore Institute of Technology                 \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
//Global variables  
char current\_Player[25];  
//char winner\_Player[25];//used to see which player won  
int end\_flag=0;//used in end\_game() function  
int help\_flag=0;//used in end\_game() function  
  
struct token{  
int x, y;// x and y coordinate  
int king\_flag; //for whoever king\_flag=1, that token is king  
};  
//struct token token\_X, token\_O;  
  
struct player{  
int count;//count for score  
int current\_flag;//for whoever current\_flag=1, that is current player  
char name[25];//name of player  
};  
  
struct player player\_X, player\_O;  
  
 char checkers[8][8]={  
    {'O',' ','O',' ','O',' ','O',' '},  
    {' ','O',' ','O',' ','O',' ','O'},  
    {'O',' ','O',' ','O',' ','O',' '},  
    {' ',' ',' ',' ',' ',' ',' ',' '},  
    {' ',' ',' ',' ',' ',' ',' ',' '},  
    {' ','X',' ','X',' ','X',' ','X'},  
    {'X',' ','X',' ','X',' ','X',' '},  
    {' ','X',' ','X',' ','X',' ','X'}  
    };  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of declaration of Global variables \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Start of colored text functions \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//red color  
void red(){  
    printf("\033[1;31m");  
}  
void reset();  
  
  
  
//green color  
void green(){  
    printf("\033[1;32m");  
}  
void reset();  
  
  
  
//yellow color  
void yellow(){  
    printf("\033[1;33m");  
}  
void reset();  
  
  
  
//blue color  
void blue(){  
    printf("\033[1;34m");  
}  
void reset();  
  
  
  
//purple color  
void purple(){  
    printf("\033[1;35m");  
}  
void reset();  
  
  
  
//cyan color  
void cyan(){  
    printf("\033[1;36m");  
}  
void reset();  
  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of colored text functions \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* START of IMPORTANT game functions \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//function to clear screen. Use cls for TurboC and clear for GCC/G++  
void screen\_clear(void){  
     system("cls");  
 }  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of screen\_clear() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//9 9 is surrender code  
int end\_game(struct token old){  
  
    if(old.x==9 && old.y==9)  
    {  
        end\_flag=1;  
        screen\_clear();  
  
        red();  
        printf("\nGAME OVER!\n");  
        //reset();  
  
        cyan();  
        if(strcmp(current\_Player,player\_X.name)==0)  
        {  
            printf("\n%s WON!\n",player\_O.name);  
            printf("\nScore of %s is %d\n",player\_O.name, player\_O.count);  
        }  
        else  
        {  
            printf("\n%s WON!\n",player\_X.name);  
            printf("\nScore of %s is %d\n",player\_X.name, player\_X.count);  
        }  
    }  
    return end\_flag;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of end\_game() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
//8 8 is help code  
int help(struct token old){  
  
    if(old.x==8 && old.y==8)  
    {  
        help\_flag=1;  
        yellow();  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.");  
    printf("\n6. Row number and column number will be displayed on all 4 corners of the board. Enter coordinates accordingly went prompted.");  
    printf("\n7. Enter 9 9 as token coordinates to surrender.\n8. Enter 99 99 as token coordinates for help menu.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    }  
    return help\_flag;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of help() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
//coding for checker board and tokens using user-defined function  
void draw\_board(void)  
{  
  
    int i,j;  
        printf("\n    0   1   2   3   4   5   6   7 \n");  
        printf("   --- --- --- --- --- --- --- --- ");  
        for(i=0;i<8;i++)  
        {  
            printf(" \n%d ",i);  
            for(j=0;j<8;j++)  
            {  
                printf("| %c ",checkers[i][j]);  
            }  
            printf("| %d \n",i);  
            if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)  
            printf("   --- --- --- --- --- --- --- --- ");  
        }  
        printf("\n    0   1   2   3   4   5   6   7 \n");  
}  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of draw\_board() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
 void move\_O(struct token old, struct token new)//for player2  
{  
   checkers[new.y][new.x]=checkers[old.y][old.x];  
   checkers[old.y][old.x]=' ';  
  
   printf("\ncheckers[%d][%d]=%c \n",new.y,new.x,checkers[new.y][new.x]);  
   printf("checkers[%d][%d]=%c \n",old.y,old.x, checkers[old.y][old.x]);  
  
    strcpy(current\_Player,player\_O.name);  
   if(end\_game(old)==0)  
    {  
        green();  
        draw\_board();  
   }  
  
   if(checkers[new.y][new.x]=='X')  
    player\_O.count++;  
  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_O() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
void move\_X(struct token old, struct token new)//for player1  
{  
    checkers[new.y][new.x]=checkers[old.y][old.x];  
    checkers[old.y][old.x]=' ';  
  
   printf("\ncheckers[%d][%d]=%c \n",new.y,new.x,checkers[new.y][new.x]);  
   printf("checkers[%d][%d]=%c \n",old.y,old.x, checkers[old.y][old.x]);  
  
    strcpy(current\_Player,player\_X.name);  
    if(end\_game(old)==0)  
    {  
       green();  
       draw\_board();  
    }  
  
    if(checkers[new.y][new.x]=='O')  
    player\_X.count++;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of move\_X() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int get\_turn(int Turn, struct player player\_X, struct player player\_O){  
  
    struct token old, new;  
    int rem =0;  
    for(Turn=1; ;Turn++)  
    {  
        if(end\_flag==1)  
            break;  
  
        if(Turn%2==0)  
        {  
            purple();  
            printf("\n %s's Turn (O's Turn)",player\_O.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.x ,&old.y);  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.x ,&new.y);  
  
                if(help(old))  
                {  
                    //help(old);  
                    help\_flag=0;  
                    purple();  
                    printf("\n %s's Turn (O's Turn)",player\_O.name);  
                    printf("\n Write x and y coordinates of token to be moved: ");  
                    scanf("%d %d" ,&old.y ,&old.x);  
                    printf("\n Write x and y coordinates of square where token is to be placed: ");  
                    scanf("%d %d" ,&new.y ,&new.x);  
                    move\_O(old, new);  
                }  
            move\_O(old, new);  
        }  
        else  
        {  
            purple();  
            printf("\n %s's Turn (X's Turn)",player\_X.name);  
            printf("\n Write x and y coordinates of token to be moved: ");  
            scanf("%d %d" ,&old.y ,&old.x);  
  
            printf("\n Write x and y coordinates of square where token is to be placed: ");  
            scanf("%d %d" ,&new.y ,&new.x);  
  
                if(help(old))  
                {  
                    //help(old);  
                    help\_flag=0;  
                    purple();  
                    printf("\n %s's Turn (X's Turn)",player\_X.name);  
                    printf("\n Write x and y coordinates of token to be moved: ");  
                    scanf("%d %d" ,&old.y ,&old.x); //player enters coordinates of x and y, but in 2-D array it's reverse  
  
                    printf("\n Write x and y coordinates of square where token is to be placed: ");  
                    scanf("%d %d" ,&new.y ,&new.x);  
                    move\_X(old, new);  
  
                }  
            move\_X(old, new);  
            rem = 1;  
        }  
    }  
    return rem;  
}  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* End of get\_turn() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* END of IMPORTANT game functions \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 \* Start of main() function \*  
 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
  
  
int main() {  
    //char player1[25], player2[25];  
  
    int turn=2;  
  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to Start Game\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    getchar();  
    screen\_clear();  
  
    cyan();  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    printf("\nEnter name of player 1:");  
    scanf("%s", player\_X.name);  
  
    printf("Enter name of player 2:");  
    scanf("%s", player\_O.name);  
  
    getchar();  
    screen\_clear();  
  
    red();  
    printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);  
    printf("\n");  
  
    blue();  
    printf("\n# Objective: Capture Opponent's token");  
    printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");  
    printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");  
    printf("\n# Points: Number of tokens cut by the player is the player's score.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
  
    green();  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
    printf("\n1. Tokens move only along the diagonal squares in forward direction.");  
    printf("\n2. Tokens move only one square during each turn.");  
    printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");  
    printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8\*8 square board.");  
    printf("\n6. Row number and column number will be displayed on all 4 corners of the board. Enter coordinates accordingly went prompted.");  
    printf("\n7. Enter 9 9 as token coordinates to surrender.\n8. Enter 99 99 as token coordinates for help menu.\n");  
    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");  
  
    getchar();  
    screen\_clear();  
  
    draw\_board();  
    turn = get\_turn(turn, player\_X, player\_O);  
  
    return 0;  
}

Progress #11

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Welcome to draught programed by Samuela, Sakthi, Merlin, and Deepiga. \*

\* \*

\* AI & DS Department \*

\* Coimbatore Institute of Technology \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//Global variables

char current\_Player[25];

int end\_flag=0;//used in end\_game() function

int help\_flag=0;//used in help() function

struct token{

int row, col;// y-axis and x-axis coordinate

int king\_flag; //for whoever king\_flag=1, that token is king

};

struct player{

int count;//count for score

int current\_flag;//for whoever current\_flag=1, that is current player

char name[25];//name of player

};

struct player player\_X, player\_O;

char checkers[8][8]={

{'O',' ','O',' ','O',' ','O',' '},

{' ','O',' ','O',' ','O',' ','O'},

{'O',' ','O',' ','O',' ','O',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','X',' ','X',' ','X',' ','X'},

{'X',' ','X',' ','X',' ','X',' '},

{' ','X',' ','X',' ','X',' ','X'}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of declaration of Global variables \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start of colored text functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//red color

void red(){

printf("\033[1;31m");

}

void reset();

//green color

void green(){

printf("\033[1;32m");

}

void reset();

//yellow color

void yellow(){

printf("\033[1;33m");

}

void reset();

//blue color

void blue(){

printf("\033[1;34m");

}

void reset();

//purple color

void purple(){

printf("\033[1;35m");

}

void reset();

//cyan color

void cyan(){

printf("\033[1;36m");

}

void reset();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of colored text functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* START of IMPORTANT game functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//function to clear screen. Use cls for TurboC and clear for GCC/G++

void screen\_clear(void){

system("cls");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of screen\_clear() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//ends game and program exits with exit code

int end\_game(struct token old){

//9 9 is surrender code

if(old.row==9 && old.col==9)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

if(strcmp(current\_Player,player\_X.name)==0)

{

printf("\n%s WON!\n",player\_O.name);

printf("\nScore of %s is %d\n",player\_O.name, player\_O.count);

}

else

{

printf("\n%s WON!\n",player\_X.name);

printf("\nScore of %s is %d\n",player\_X.name, player\_X.count);

}

}

//player 2 captured all X tokens, so game over

else if(player\_O.count==12)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

printf("\n%s WON!\n",player\_O.name);

printf("\nScore of %s is %d\n",player\_O.name, player\_O.count);

}

//player 1 captured all O tokens, so game over

else if(player\_X.count==12)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

printf("\n%s WON!\n",player\_X.name);

printf("\nScore of %s is %d\n",player\_X.name, player\_X.count);

}

return end\_flag;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of end\_game() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//8 8 is help code

int help(struct token old){

if(old.row==8 && old.col==8)

{

help\_flag=1;

yellow();

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8x8 square board.");

printf("\n6. Row number with prefix r and column number with prefix c will be displayed on all 4 corners of the board.\n Enter coordinates accordingly when prompted.");

printf("\n7. Enter 9 9 as token coordinates to surrender.\n8. Enter 8 8 as token coordinates for help menu.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

return help\_flag;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of help() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//coding for checker board and tokens using user-defined function

void draw\_board(void)

{

int i,j;

printf("\n c0 c1 c2 c3 c4 c5 c6 c7 \n");

printf(" --- --- --- --- --- --- --- --- ");

for(i=0;i<8;i++)

{

printf(" \nr%d ",i);

for(j=0;j<8;j++)

{

printf("| %c ",checkers[i][j]);

}

printf("| r%d \n",i);

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

printf(" --- --- --- --- --- --- --- --- ");

}

printf("\n c0 c1 c2 c3 c4 c5 c6 c7 \n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of draw\_board() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void move\_O(struct token old, struct token new)//for player2

{

checkers[new.row][new.col]=checkers[old.row][old.col];

checkers[old.row][old.col]=' ';

printf("\n new checkers[%d][%d]=%c \n",new.row,new.col,checkers[new.row][new.col]);

printf("old checkers[%d][%d]=%c \n",old.row,old.col, checkers[old.row][old.col]);

strcpy(current\_Player,player\_O.name);

if(end\_game(old)==0)

{

green();

draw\_board();

}

if(checkers[new.row][new.col]=='X')

player\_O.count++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of move\_O() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void move\_X(struct token old, struct token new)//for player1

{

checkers[new.row][new.col]=checkers[old.row][old.col];

checkers[old.row][old.col]=' ';

printf("\n new checkers[%d][%d]=%c \n",new.row,new.col,checkers[new.row][new.col]);

printf("old checkers[%d][%d]=%c \n",old.row,old.col, checkers[old.row][old.col]);

strcpy(current\_Player,player\_X.name);

if(end\_game(old)==0)

{

green();

draw\_board();

}

if(checkers[new.row][new.col]=='O')

player\_X.count++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of move\_X() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int get\_turn(int Turn, struct player player\_X, struct player player\_O){

struct token old, new;

int rem =0;

for(Turn=1; ;Turn++)

{

if(end\_flag==1)

break;

if(Turn%2==0)

{

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

if(help(old))

{

//help(old);

help\_flag=0;

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_O(old, new);

}

move\_O(old, new);

}

else

{

purple();

printf("\n %s's Turn (X's Turn)",player\_X.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

if(help(old))

{

help\_flag=0;

purple();

printf("\n %s's Turn (X's Turn)",player\_X.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_X(old, new);

}

move\_X(old, new);

rem = 1;

}

}

return rem;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of get\_turn() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* END of IMPORTANT game functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start of main() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main() {

//char player1[25], player2[25];

int turn=2;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to Start Game\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

cyan();

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player\_X.name);

printf("Enter name of player 2:");

scanf("%s", player\_O.name);

getchar();

screen\_clear();

red();

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);

printf("\n");

blue();

printf("\n# Objective: Capture Opponent's token");

printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");

printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");

printf("\n# Points: Number of tokens cut by the player is the player's score.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

green();

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

printf("\n3. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n4. Player 1's token is 'X', Player 2's token is 'O'.\n5. Each Player has 12 tokens on a 8x8 square board.");

printf("\n6. Row number with prefix r and column number with prefix c will be displayed on all 4 corners of the board.\n Enter coordinates accordingly when prompted.");

printf("\n7. Enter 9 9 as token coordinates to surrender.\n8. Enter 8 8 as token coordinates for help menu.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

draw\_board();

turn = get\_turn(turn, player\_X, player\_O);

return 0;

}

Progress #12

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Welcome to draught programed by Samuela, Sakthi, Merlin, and Deepiga. \*

\* \*

\* AI & DS Department \*

\* Coimbatore Institute of Technology \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

//Global variables

char current\_Player[25];

int end\_flag=0;//used in end\_game() function

int help\_flag=0;//used in help() function

int check\_flag\_X=0;//if it's set to 1, then move made by player 1 is invalid

int check\_flag\_O=0;//if it's set to 1, then move made by player 2 is invalid

int king\_flag=0; //for whoever king\_flag=1, that token is king

struct token{

int row, col;// y-axis and x-axis coordinate

};

struct player{

int count;//count for score

char name[25];//name of player

};

struct player player\_X, player\_O;

char checkers[8][8]={

{'O',' ','O',' ','O',' ','O',' '},

{' ','O',' ','O',' ','O',' ','O'},

{'O',' ','O',' ','O',' ','O',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ',' ',' ',' ',' ',' ',' ',' '},

{' ','X',' ','X',' ','X',' ','X'},

{'X',' ','X',' ','X',' ','X',' '},

{' ','X',' ','X',' ','X',' ','X'}

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of declaration of Global variables \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start of colored text functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//red color

void red(){

printf("\033[1;31m");

}

void reset();

//green color

void green(){

printf("\033[1;32m");

}

void reset();

//yellow color

void yellow(){

printf("\033[1;33m");

}

void reset();

//blue color

void blue(){

printf("\033[1;34m");

}

void reset();

//purple color

void purple(){

printf("\033[1;35m");

}

void reset();

//cyan color

void cyan(){

printf("\033[1;36m");

}

void reset();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of colored text functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* START of IMPORTANT game functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//function to clear screen. Use cls for TurboC and clear for GCC/G++

void screen\_clear(void){

system("cls");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of screen\_clear() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//ends game and program exits with exit code

int end\_game(struct token old){

//9 9 is surrender code

if(old.row==9 && old.col==9)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

if(strcmp(current\_Player,player\_X.name)==0)

{

printf("\n%s WON!\n",player\_O.name);

printf("\nScore of %s is %d\n",player\_O.name, player\_O.count);

}

else

{

printf("\n%s WON!\n",player\_X.name);

printf("\nScore of %s is %d\n",player\_X.name, player\_X.count);

}

}

//player 2 captured all X tokens, so game over

else if(player\_O.count==12)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

printf("\n%s WON!\n",player\_O.name);

printf("\nScore of %s is %d\n",player\_O.name, player\_O.count);

}

//player 1 captured all O tokens, so game over

else if(player\_X.count==12)

{

end\_flag=1;

screen\_clear();

red();

printf("\nGAME OVER!\n");

cyan();

printf("\n%s WON!\n",player\_X.name);

printf("\nScore of %s is %d\n",player\_X.name, player\_X.count);

}

return end\_flag;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of end\_game() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//check if king token or not

void check\_king()

{

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of check\_king() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//check if move made by token X is valid or not

int check\_valid\_X(struct token old, struct token new)

{

if(king\_flag==0)

{

//Anything other than one diagonally right movement not allowed

if(checkers[new.row][new.col]!=checkers[old.row-1][old.col+1])

{

check\_flag\_X=1;

printf("\nInvalid move! Try again.\n");

}

//Anything other than one diagonally left movement not allowed

else if(checkers[new.row][new.col]!=checkers[old.row-1][old.col-1])

{

check\_flag\_X=1;

printf("\nInvalid move! Try again.\n");

}

//player 1 has X tokens, so he can't select anything else

else if(checkers[old.row][old.col]!='X')

{

check\_flag\_X=1;

printf("\nInvalid move! Try again.\n");

}

//Token can't be placed in a square where another X token is already present

else if(checkers[new.row][new.col]=='X')

{

check\_flag\_X=1;

printf("\nInvalid move! Try again.\n");

}

//Token can't be placed outside the board

else if(new.row>7 && new.col>7)

{

check\_flag\_X=1;

printf("\nInvalid move! Try again.\n");

}

}

return check\_flag\_X;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of check\_valid\_X() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//check if move made by token O is valid or not

int check\_valid\_O(struct token old, struct token new)

{

if(king\_flag==0)

{

//Anything other than one diagonally right movement not allowed

if(checkers[new.row][new.col]!=checkers[old.row+1][old.col+1])

{

check\_flag\_O=1;

printf("\nInvalid move! Try again.\n");

}

//Anything other than one diagonally left movement not allowed

else if(checkers[new.row][new.col]!=checkers[old.row+1][old.col-1])

{

check\_flag\_O=1;

printf("\nInvalid move! Try again.\n");

}

//player 2 has O tokens, so he can't select anything else

else if(checkers[old.row][old.col]!='O')

{

check\_flag\_O=1;

printf("\nInvalid move! Try again.\n");

}

//Token can't be placed in a square where another O token is already present

else if(checkers[new.row][new.col]=='O')

{

check\_flag\_O=1;

printf("\nInvalid move! Try again.\n");

}

//Token can't be placed outside the board

else if(new.row>7 && new.col>7)

{

check\_flag\_O=1;

printf("\nInvalid move! Try again.\n");

}

}

return check\_flag\_O;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of check\_valid\_O() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//8 8 is help code

int help(struct token old){

if(old.row==8 && old.col==8)

{

help\_flag=1;

yellow();

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

printf("\n3. Once a token reaches other end of the board, it'll become king and can move in all directions,\n any number of squares during each turn.");

printf("\n4. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n5. Player 1's token is 'X', Player 2's token is 'O'.\n6. Each Player has 12 tokens on a 8x8 square board.");

printf("\n7. Row number with prefix r and column number with prefix c will be displayed on all 4 corners of the board.\n Enter coordinates accordingly when prompted.");

printf("\n8. Enter 9 9 as token coordinates to surrender.\n9. Enter 8 8 as token coordinates for help menu.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

}

return help\_flag;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of help() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//coding for checker board and tokens using user-defined function

void draw\_board(void)

{

int i,j;

printf("\n c0 c1 c2 c3 c4 c5 c6 c7 \n");

printf(" --- --- --- --- --- --- --- --- ");

for(i=0;i<8;i++)

{

printf(" \nr%d ",i);

for(j=0;j<8;j++)

{

printf("| %c ",checkers[i][j]);

}

printf("| r%d \n",i);

if(i==0||i==1||i==2||i==3||i==4||i==5||i==6||i==7)

printf(" --- --- --- --- --- --- --- --- ");

}

printf("\n c0 c1 c2 c3 c4 c5 c6 c7 \n");

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of draw\_board() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void move\_O(struct token old, struct token new)//for player2

{

checkers[new.row][new.col]=checkers[old.row][old.col];

checkers[old.row][old.col]=' ';

printf("\n new checkers[%d][%d]=%c \n",new.row,new.col,checkers[new.row][new.col]);

printf("old checkers[%d][%d]=%c \n",old.row,old.col, checkers[old.row][old.col]);

strcpy(current\_Player,player\_O.name);

if(end\_game(old)==0)

{

green();

draw\_board();

}

if(checkers[new.row][new.col]=='X')

player\_O.count++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of move\_O() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void move\_X(struct token old, struct token new)//for player1

{

checkers[new.row][new.col]=checkers[old.row][old.col];

checkers[old.row][old.col]=' ';

printf("\n new checkers[%d][%d]=%c \n",new.row,new.col,checkers[new.row][new.col]);

printf("old checkers[%d][%d]=%c \n",old.row,old.col, checkers[old.row][old.col]);

strcpy(current\_Player,player\_X.name);

if(end\_game(old)==0)

{

green();

draw\_board();

}

if(checkers[new.row][new.col]=='O')

player\_X.count++;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of move\_X() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int get\_turn(int Turn, struct player player\_X, struct player player\_O){

struct token old, new;

int rem =0;//remainder of Turn%2

for(Turn=1; ;Turn++)

{

if(end\_flag==1)

break;

if(Turn%2==0)

{

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

if(help(old))

{

//help(old);

help\_flag=0;

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_O(old, new);

}

if(check\_valid\_O(old,new))

{

check\_flag\_O=0;

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_O(old, new);

}

move\_O(old, new);

}

else

{

purple();

printf("\n %s's Turn (X's Turn)",player\_X.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

if(help(old))

{

help\_flag=0;

purple();

printf("\n %s's Turn (X's Turn)",player\_X.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_X(old, new);

}

if(check\_valid\_X(old,new))

{

check\_flag\_X=0;

purple();

printf("\n %s's Turn (O's Turn)",player\_O.name);

printf("\n Write row number and column number of token to be moved: ");

scanf("%d %d" ,&old.row ,&old.col);

printf("\n Write row number and column number of square where token is to be placed: ");

scanf("%d %d" ,&new.row ,&new.col);

move\_X(old, new);

}

move\_X(old, new);

rem = 1;

}

}

return rem;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* End of get\_turn() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* END of IMPORTANT game functions \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Start of main() function \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main() {

//char player1[25], player2[25];

int turn=2;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to Start Game\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

cyan();

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME!\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DRAUGHTS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nEnter name of player 1:");

scanf("%s", player\_X.name);

printf("Enter name of player 2:");

scanf("%s", player\_O.name);

getchar();

screen\_clear();

red();

printf("\n\*\*\*Good day %s and %s, let's start the game. All the best!!!\*\*\*\n", player\_X.name, player\_O.name);

printf("\n");

blue();

printf("\n# Objective: Capture Opponent's token");

printf("\n# Cutting of opponent's token: A player may jump one of his tokens onto one square occupied by his opponent's token.");

printf("\n# Captured token: The opponent's token is cut by the player's token and removed from the board.");

printf("\n# Points: Number of tokens cut by the player is the player's score.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

green();

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*RULES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Please READ carefully\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1. Tokens move only along the diagonal squares in forward direction.");

printf("\n2. Tokens move only one square during each turn.");

printf("\n3. Once a token reaches other end of the board, it'll become king and can move in all directions,\n any number of squares during each turn.");

printf("\n4. You can continue the game until you capture all opponent tokens or until you surrender.");

printf("\n5. Player 1's token is 'X', Player 2's token is 'O'.\n6. Each Player has 12 tokens on a 8x8 square board.");

printf("\n7. Row number with prefix r and column number with prefix c will be displayed on all 4 corners of the board.\n Enter coordinates accordingly when prompted.");

printf("\n8. Enter 9 9 as token coordinates to surrender.\n9. Enter 8 8 as token coordinates for help menu.\n");

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*Press ENTER key to continue\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

getchar();

screen\_clear();

draw\_board();

turn = get\_turn(turn, player\_X, player\_O);

return 0;

}

See next pdf

BIBLIOGRAPY

<https://cboard.cprogramming.com/c-programming/177496-checkers-game-finally-finished.html>

https://www.youtube.com/watch?v=Z\_f6yCvAZfQ