



```
#include<conio.h>
#include<fstream.h>
#include<stdio.h>
#include<dos.h>
#include<string.h>
#include<iomanip.h>
class chess_game
{
int game_no;
char white_name[30];
char black_name[30];
char moves[100][3];
int moves_no;
int winner; //1 for white and 2 for black
public:
void manual(); //to view instructions and openings
void openings(); //to view the openings our program displays
void main_menu(); //to display the main menu
void add(); //to add a new chess game
void check_winner(); //to check winner of game
void display(); //to display the chess game
void all_games(); //to display recorded games
void edit(); //to edit the game record
int check(int); //to check game status
void opening_checker(); //to check which opening has been played
void modify(int); //to modify the record
void delete_rec(int); //to delete the record
};
void chess_game::main_menu()
{
int choice;
while(choice!=6)
{
clrscr();
cout<<"\n\t\t\t *****";
cout<<"\n\t\t\t * MAIN MENU *";
```

```

cout<<"\n\t\t\t * CHESS GAME STORAGE *";
cout<<"\n\t\t\t *****";
cout<<"\n\n\n\t\t\t1.User Manual";
cout<<"\n\t\t\t2.Enter A Game";
cout<<"\n\t\t\t3.View And Analyze A Game";
cout<<"\n\t\t\t4.View All Recorded Games";
cout<<"\n\t\t\t5.Modify Game Details";
cout<<"\n\t\t\t6.Exit";
cout<<"\n\n\n\t\t\tEnter Your Choice: ";
cin>>choice;
switch(choice)
{
case 1: manual();
break;
case 2: add();
break;
case 3: display();
break;
case 4: all_games();
break;
case 5: edit();
break;
case 6: break;
default:
{
cout<<"\n\n\n\t\t\tWrong choice!!!";
cout<<"\n\t\t\tPress any key to continue!!";
getch();
main_menu();
}
}
}
}

void chess_game::manual()
{
char n;
clrscr();
cout<<"\n Important Instructions ";
cout<<"\n *Each move should consist of two characters and a number(eg. ne4)";

```

```

cout<<"\n *Use initial letter of piece to signify it(eg. use p for pawn)";
cout<<"\n *Seperate black and white moves by pressing the enter button";
cout<<"\n *The last move should end with a # for checkmate";
cout<<"\n\n Would you like to view the openings the program processes?(y/n) \n ";
cin>>n;
if(n=='y'){
openings();
}
cout<<"\n\n Press any key to continue!!";
getch();
}
void chess_game::openings()
{
cout<<"\n These are the openings that our program recognises";
cout<<"\n 1.Sicilian Defence 1)pe4 pc5";
cout<<"\n 2.Nimzowitsch Defence 1)pe4 nc6";
cout<<"\n 3.French Defence 1)pe4 pe6";
cout<<"\n 4.Lopez Opening 1)pe4 pe5 2)pc3";
cout<<"\n 5.Owen's Defence 1)pe4 pb6";
cout<<"\n 6.Bishop's Opening 1)pe4 pe5 2)bc4 ";
cout<<"\n 7.Queen's Gambit 1)pd4 pd5 2)pc4";
cout<<"\n 8.Dutch Defence 1)pd4 pf5";
cout<<"\n 9.Stonewall Attack 1)pd4 pd5 2)pe3";
cout<<"\n 10.Indian Defence 1)pd4 nf6";
}
void chess_game::opening_checker(){
if(moves[0][0]=='p'){
if(moves[0][1]=='e'){
if(moves[0][2]=='4'){
if(moves[0][3]=='p'){
if(moves[0][4]=='c'){
if(moves[0][5]=='5'){
cout<<"Sicillian Defence";
}
}
}
else if(moves[0][4]=='e'){
if(moves[0][5]=='6'){
cout<<"French Defence";
}
}
}
}
}
}
}
}
}
}
}

```

```
else if(moves[0][5]=='5'){
if(moves[0][6]=='p'){
if(moves[0][7]=='c'){
if(moves[0][8]=='3'){
cout<<"Lopez Opening";
}
}
}
else if(moves[0][6]=='b'){
if(moves[0][7]=='c'){
if(moves[0][8]=='4'){
cout<<"Bishop's Opening";
}
}
}
}
else if(moves[0][4]=='b'){
if(moves[0][5]=='6'){
cout<<"Owen's Defence";
}
}
}
else if(moves[0][3]=='n'){
if(moves[0][4]=='c'){
if(moves[0][5]=='6'){
cout<<"Nimzowitsch Defence";
}
}
}
}
else if(moves[0][1]=='d'){
if(moves[0][2]=='4'){
if(moves[0][3]=='p'){
if(moves[0][4]=='f'){
if(moves[0][5]=='5'){
cout<<"Dutch Defence";
}
}
}
}
}
```

```

}
else if(moves[0][4]=='d'){
if(moves[0][5]=='5'){
if(moves[0][6]=='p'){
if(moves[0][7]=='c'){
if(moves[0][8]=='4'){
cout<<"Queen's Gambit";
}
}
else if(moves[0][7]=='e'){
if(moves[0][8]=='3'){
cout<<"Stonewall Attack";
}
}
}
}
}
}
else if(moves[0][3]=='n'){
if(moves[0][4]=='c'){
if(moves[0][5]=='6'){
cout<<"Indian Defence";
}
}
}
}
}
}
else{
cout<<"None recorded";
}
}
void chess_game::add()
{
clrscr();
int r,flag;
ofstream fout("Record.dat",ios::app|ios::binary);
cout<<"\n Enter Chess Game Details";
cout<<"\n *****";

```

```
cout<<"\n\n Chess game number: ";
cin>>r;
flag=check(r);
if(flag==1){
cout<<"\n This game is already recorded";
cout<<"\n Please enter a different game";
}
else
{
game_no=r;
cout<<" Name of white player: ";
gets(white_name);
cout<<" Name of black player: ";
gets(black_name);
cout<<" Refer to the user manual to see how to input moves";
cout<<"\n Enter the moves played: ";
int i=0;
int j=1;
while(strcmp(moves[i-1],"#")!=0){
cout << j << " ";
gets(moves[i]);
cout << " ";
moves_no=j;
if(i%2==1){
j++;
}
i++;
}
if(i%2==1){
//black won
winner=2;
moves_no--;
}
else{
//white won
winner=1;
}
fout.write((char*)this,sizeof(chess_game));
```

```

cout<<"\n Game was successfully recorded!!!";
}
cout<<"\n\n Press any key to continue!!";
getch();
fout.close();
}
void chess_game::display()
{
clrscr();
ifstream fin("Record.dat",ios::in);
int r,flag;
cout<<"\n Enter chess game number: ";
cin>>r;
while(!fin.eof())
{
fin.read((char*)this,sizeof(chess_game));
if(game_no==r)
{
clrscr();
cout<<"\n Game Details";
cout<<"\n *****";
cout<<"\n\n Game no: "<<game_no;
cout<<"\n Name of opening played: "; opening_checker();
cout<<"\n Name of white player: "<<white_name;
cout<<"\n Name of black player: "<<black_name;
cout<<"\n Winner: "; check_winner();
cout<<"\n Number of Moves: "<<moves_no;
cout<<"\n Moves: "<<moves[0];
flag=1;
break;
}
}
if(flag==0){
cout<<"\n Sorry this game is not registered yet!!";
}
cout<<"\n\n Press any key to continue!!";
getch();
fin.close();
}

```

```

void chess_game::check_winner(){
if(winner==1){
cout<<"White";
}
else{
cout<<"Black";
}
}
void chess_game::all_games()
{
clrscr();
ifstream fin("Record.dat",ios::in);
cout<<"\n\t\t\tList Of Recorded Games";
cout<<"\n\t\t\t*****";
cout<<"\n\n Game No.\tWhite Player\t\tBlack Player\t\tWinner\n";
while(!fin.eof())
{
fin.read((char*)this,sizeof(chess_game));
cout<<"\n\n "<<game_no<<"\t\t"<<white_name;
cout<<"\t\t"<<black_name<<"\t\t";check_winner();
}
cout<<"\n\n\n\t\t\tPress any key to continue!!";
getch();
fin.close();
}
void chess_game::edit()
{
clrscr();
int choice,r;
cout<<"\n EDIT MENU";
cout<<"\n *****";
cout<<"\n\n 1.Modify Chess Game Record";
cout<<"\n 2.Delete Chess Game Record";
cout<<"\n Enter your choice: ";
cin>>choice;
clrscr();
cout<<"\n Enter chess game number: ";
cin>>r;

```



```

switch(choice)
{
case 1: modify(r);
break;
case 2: delete_rec(r);
break;
default: cout<<"\n Wrong Choice!!";
}
cout<<"\n Press any key to continue!!!";
getch();
}
int chess_game::check(int r)
{
int flag=0;
ifstream fin("Record.dat",ios::in);
while(!fin.eof())
{
fin.read((char*)this,sizeof(chess_game));
if(game_no==r)
{
flag=1;
break;
}
}
fin.close();
return(flag);
}
void chess_game::modify(int r)
{
long pos,flag=0;
fstream file("Record.dat",ios::in|ios::out);
while(!file.eof())
{
pos=file.tellg();
file.read((char*)this,sizeof(chess_game));
if(game_no==r)
{
cout<<"\n Enter New Details";
cout<<"\n *****";
}
}
}

```

```

cout<<"\n Name of white player: ";
gets(white_name);
cout<<" Name of black player: ";
gets(black_name);
cout<<" Refer to the user manual to see how to input moves";
cout<<"\n Enter the moves played: ";
int i=0;
int j=1;
while(strcmp(moves[i-1],"#")!=0){
cout << j << ". ";
gets(moves[i]);
cout << " ";
moves_no=j;
if(i%2==1){
j++;
}
i++;
}
if(i%2==1){
//black won
winner=2;
moves_no--;
}
else{
//white won
winner=1;
}
file.seekg(pos);
file.write((char*)this,sizeof(chess_game));
cout<<"\n Record is modified!!!";
flag=1;
break;
}
}
if(flag==0)
cout<<"\n Sorry this game is not registered yet!!!";
file.close();
}

```

```

void chess_game::delete_rec(int r)
{
int flag=0;
char ch;
ifstream fin("Record.dat",ios::in);
ofstream fout("temp.dat",ios::app);
while(!fin.eof())
{
fin.read((char*)this,sizeof(chess_game));
if(game_no==r)
{
cout<<"\n Name of white player: "<<white_name;
cout<<"\n Name of black player: "<<black_name;
cout<<"\n Winner: ";check_winner();
cout<<"\n\n Do you want to delete this record(y/n): ";
cin>>ch;
if(ch=='n')
fout.write((char*)this,sizeof(chess_game));
flag=1;
}
else
fout.write((char*)this,sizeof(chess_game));
}
fin.close();
fout.close();
if(flag==0)
cout<<"\n Sorry this game is not registered yet!!!";
else
{
remove("Record.dat");
rename("temp.dat","Record.dat");
}
}
void main()
{
chess_game h;
textmode(C80);
textbackground(WHITE);
textcolor(BLUE);

```

