War/I Declare War
Card Game

Laurie Guimont

CSC 5

Summer 2016

45276

TABLE OF CONTENTS

INTRODUCTION 3

HOW THE CARD GAME WORKS 3

OBJECT OF THE GAME 3
RULES OF THE GAME 3
HOW TO WAGE WAR 3

MY APPROACH TO THE GAME 4

TRANSLATING GAME PLAY RULES TO PROGRAMMING LANGUAGE 4
SIMILARITIES TO THE CARD GAME 4
DIFFERENCES FROM THE CARD GAME 5

THE LOGIC OF IT ALL 5

FLOWCHART 5

CONSTRUCTS & CONCEPTS UTILIZED 9

IOSTREAM LIBRARY 9
CSTDLIB LIBRARY 10
CTIME LIBRARY 10
IOMANIP LIBRARY 10
STRING LIBRARY 10
CMATH LIBRARY 10
FSTREAM LIBRARY 10
DATA TYPES 11
CONDITIONAL STATEMENTS 11
LOOPS 11

PROOF OF A WORKING PRODUCT 11

REFERENCES 15

PROGRAM 15

Introduction

"WAR! huh! Yeah,
What is it good for? Absolutely…"
–Edwin Starr

…any good pass time with friends and family (the card game, that is).

War, also referred to by some as I Declare War, is a popular 2-Player standard card game. The game is played by people of all ages, and can serve as a useful way of helping young children learn how to count. All the players are required to do is display the cards they are dealt and compare them. Although the game does not require much logic and may be considered simplistic to some, the game is still extremely competitive if the right cards are dealt in the right order, or in other words, in an order that helps players beat their opponent. This dynamic of having cards randomly arranged in a way that benefits either player appealed to me and inspired me to write a program around the game.

How the Card Game Works

Object of the Game

To accumulate all 52 cards.

Rules of the Game

War is typically a two person game. The game is very simple:

1. Shuffle and deal the cards evenly between the two players. Therefore, each player should have 26 cards. Jokers are not used in this game.
2. Players should then turn over the top cards in their pile at the same time. Whoever has the higher value card wins both cards. The ranks of cards are as follows:
    - All number cards are valued according to their number.
    - Of the face cards the Ace is the highest overall card, followed by the King, then Queen, and the Jack is the lowest ranked face card. Face cards beat number cards.
3. Keep playing until one of the players has collected all of the cards in the deck.

How to Wage War

If the players turn over cards that have the same card value, war is waged! At this point, both players must place 2 to 4 cards faced down, then turn over the proceeding card. Whichever

player has the higher war card gets all the cards put down, including the cards faced down and the cards that initiated the war.

> Note: The number of cards placed faced down before overturning one is based upon player preference. I have seen games played where only 2 cards were placed faced down, but have also seen games where 4 cards were placed down. The latter is interesting because while placing your cards down, the players count and say aloud, "1, 2, 3, 4" then proceed with "I declare war" while overturning the fifth card at the same time the word "war" is said.
>
> Note: More than one war can be declared in a round. If players throw the same war card down, another round of "faced down" cards must be placed along with another war card. This process should be repeated until one of the players has a higher war card.

My Approach to the Game

Translating Game Play Rules to Programming Language

While thinking about how I was going to program this game, a couple questions arose:

- "Since the card game has four suites, meaning four of each card, how do I tell the computer that I want to limit the number of times a random number is chosen?"
- "Should I have the computer 'deal' 26 cards to the user/player, and then have the player chose from their 'hand'?"
- "How will a player win or lose the game?"

After a couple of hours of planning my program and toiling with the above questions, I did some research and found that most (if not all) of the questions I was asking myself had a common answer: arrays. Well, I hadn't learned arrays at that point. Therefore, I had to come up with a way to cope with these problems using the constructs and concepts that I already knew.

Similarities to the Card Game

My War program follows the same rules of play as the card game:

- The user and the computer "throw down" a card, then it is determined who has the higher card
- If the same card is thrown down, both user and computer place cards faced down, then reveal the war card.
    - This is repeated if the war cards are the same.

Differences from the Card Game

The main difference from the card game is the score. I decided that in order to determine the winner of each round in the game, I would numerically value each card and add up points.

- If you win, you gain the value of the card you put down as well as the value of the card the computer put down.
- If you lose, you lose the value of the card you put down. The same goes for a computer loss.

Since the face cards don't have a numeric value in the regular card game, I assigned a value to each according to their rank in the game. Therefore, the Jack is valued at 11 points and the Ace is valued at 14 points, with the Queen at 12 points, and the King at 13.

In response to my concern about limiting the number of times a card is chosen, I decided to not worry about that and treat the game as if two player were not being dealt cards, but instead were drawing them and putting them back in the pile, all while keeping score. It's like "War, with Replacements."

Finally, a player wins or loses based on their score at the time they decide to finish the game. If the player is tired of playing after a while, they can exit the game and the computer will tell them their final score. If the score is higher than the computer's, they have one. If not, they have lost.

The Logic of it All

Flowchart

Since my flowchart is extremely long, I will break it up into smaller pieces and accompany it with pseudocode here. To view my complete flowchart, please visit: http://www.gliffy.com/go/publish/10930307

Put in opening comments

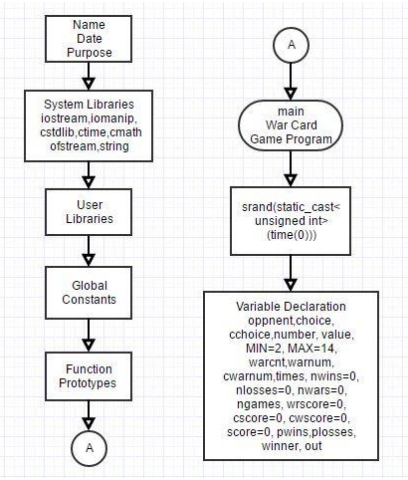
Bring in system libraries

Enter main, then immediately

set random number seed

Declare all variables, initiate some

now and some later.



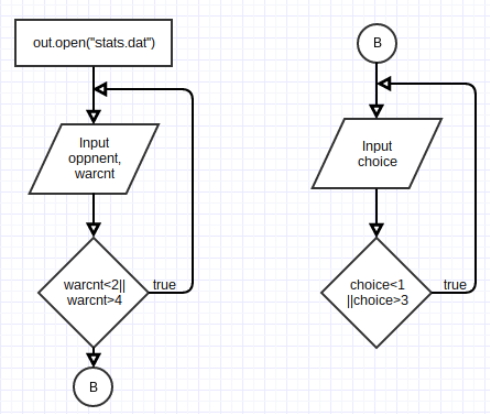
Open file that we will write data to

Input opponent name and number of faced down cards

Loop this step until valid data is inputted.

Input valid menu choice.

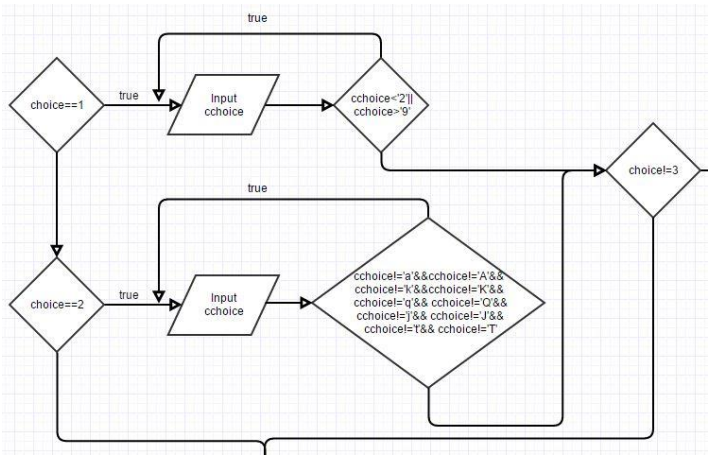
Verify valid data with while loop again



If "number card" or "face card",

input card choice

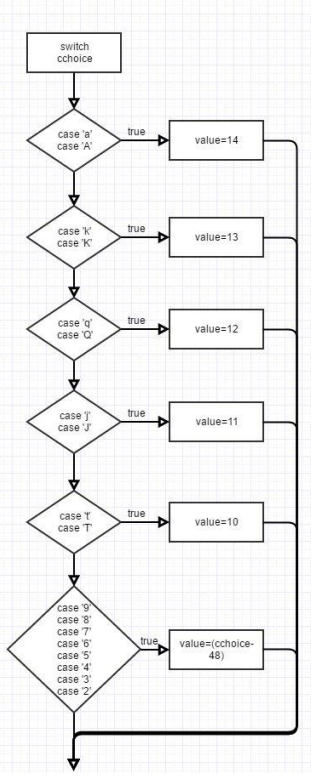
Verify that all data is valid.



As long as "End Menu" is not chosen,

computer processes input by applying

assigned value in switch statement



Computer chooses a random number and

compares number to input value

If value is bigger than random number

User wins round and

Score is calculated

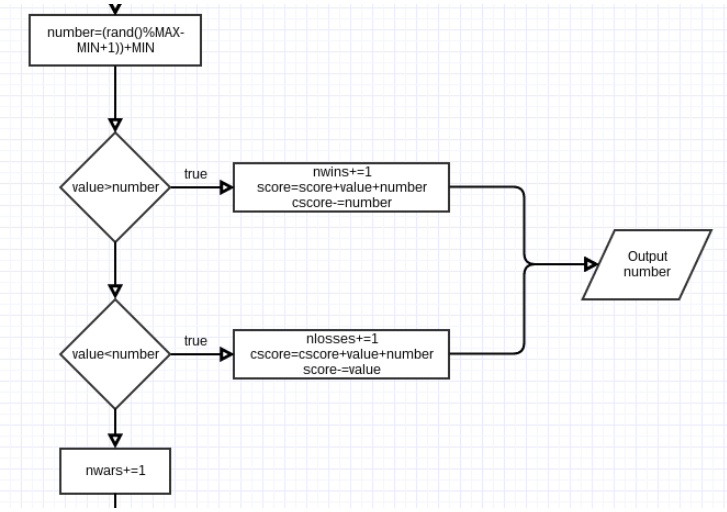
If value is smaller, computer wins.

Computer wins round

Score is calculated

If both numbers are equal

User has entered war



Prompt user to enter face down cards

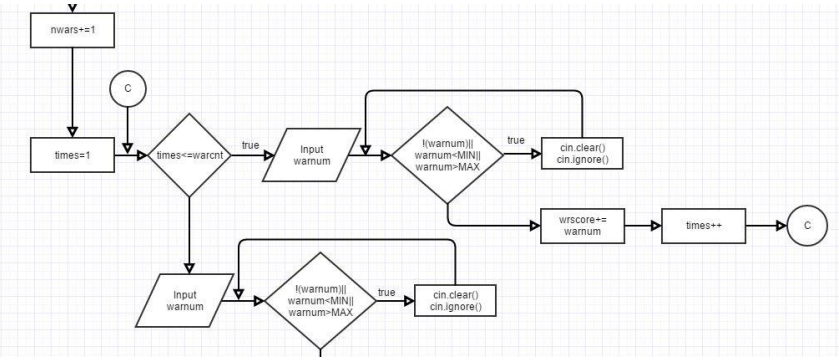
According to initial input at start

of the game

Validate

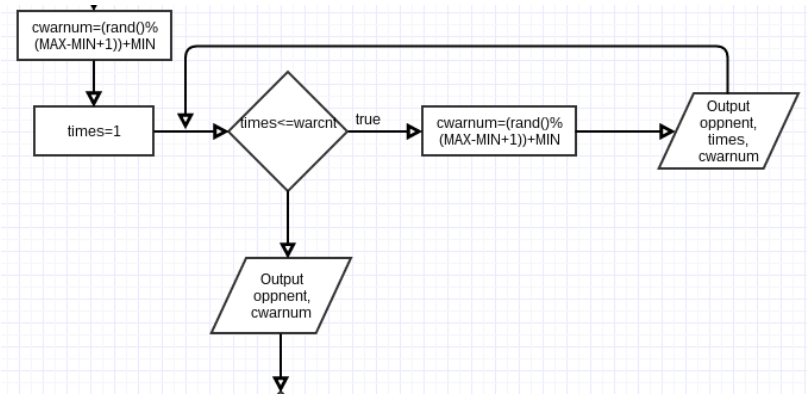
Prompt user to enter war card

Validate



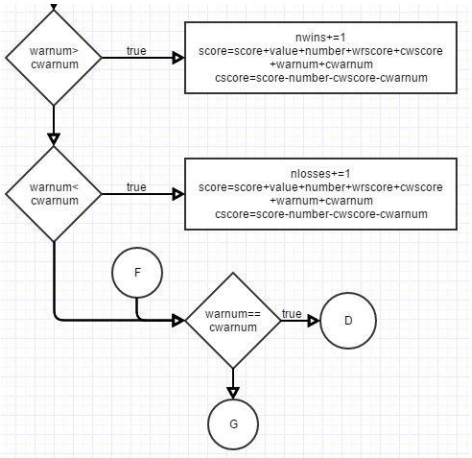
Computer now chooses faced down cards

Computer chooses war card

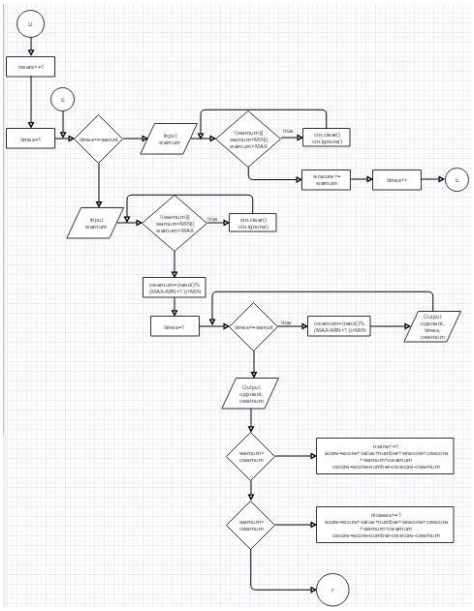


War Card comparison is made

whoever has the higher number, win.

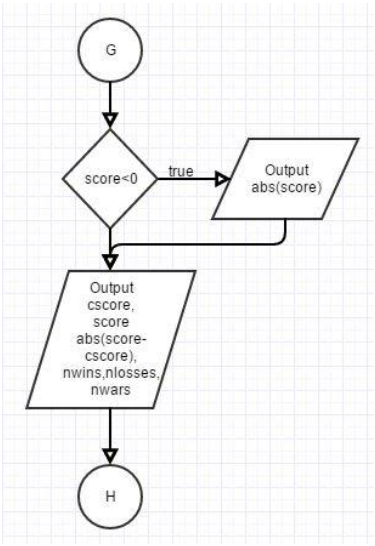


If equal, repeat war.



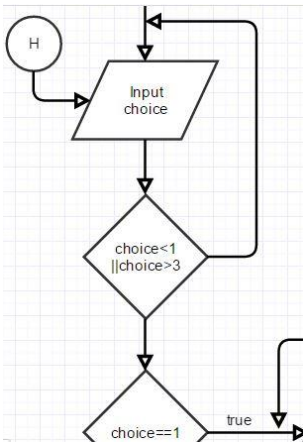
Game stats are kept during gameplay.

Display after each round



Return to initial choice and repeat entire process

as long as "End Program"(3) is not selected.

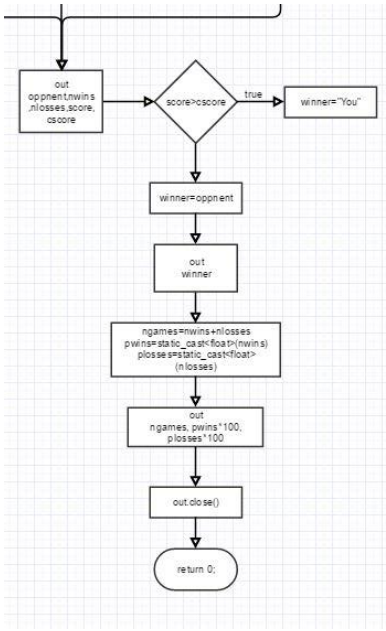


If 3 is selected, thank user for playing and write

finishing stats to the output file.

Calculate some statistics then close the file

Return 0; the program is complete



Constructs & Concepts Utilized

iostream Library

Name	Frequency	Description	Location
static_cast	3	Statically cast as different variable	Line 44,192,193
cout	54	Output Data	Throughout
cin	12	Input Data	Throughout
getline()	1	Reads string data	Line 66
cin.ignore()	2	Prevented input problems	Line 283,326
cin.clear()	2	Stopped infinite loop	Line 282,325

cstdlib Library

Name	Frequency	Description	Location
srand()	1	Random # seed	Line 44
rand()	3	Generates rand #	Line 83,271,303

ctime Library

Name	Frequency	Description	Location
time	1	Set current time	Line 44

iomanip Library

Name	Frequency	Description	Location
fixed	1	Format final game stats	Line 196
setprecision()	1	Format final game stats	Line 196
showpoint	1	Format final game stats	Line 196
setw()	12	Format final game stats	Line 174,175, 177,178,240,247, 254,255,256,264, 265,266

string Library

Name	Frequency	Description	Location
string	8	Declare var./parameters	Line 33,34,35 46,58,252,269,299
getline()	already mentioned	already mentioned	already mentioned

cmath Library

Name	Frequency	Description	Location
abs()	2	Neg. Score Alert Point Difference	Line 256,260

fstream Library

Name	Frequency	Description	Location
out.open()	1	Open file	Line 62
out.close()	1	Close file	Line 201
in.open()	1	Open file	Line 230

in.close()	1	Close file	Line 233
out	12	Write to file	Line 172-178, 185,189,196-198
in	1	Read in file	Line 233
ofstream	1	Declare var.	Line 59
ifstream	1	Declare var.	Line 228

Data Types:

Data Types	Frequency	Location
int	11	Line 32,33,36,37,40
unsigned int	7	Line 27,30,38,39
char	1	Line 31
string	2	Line 29,42
float	2	Line 41
ofstream	1	Line 43
bool	1	Line 207

Conditional Statements:

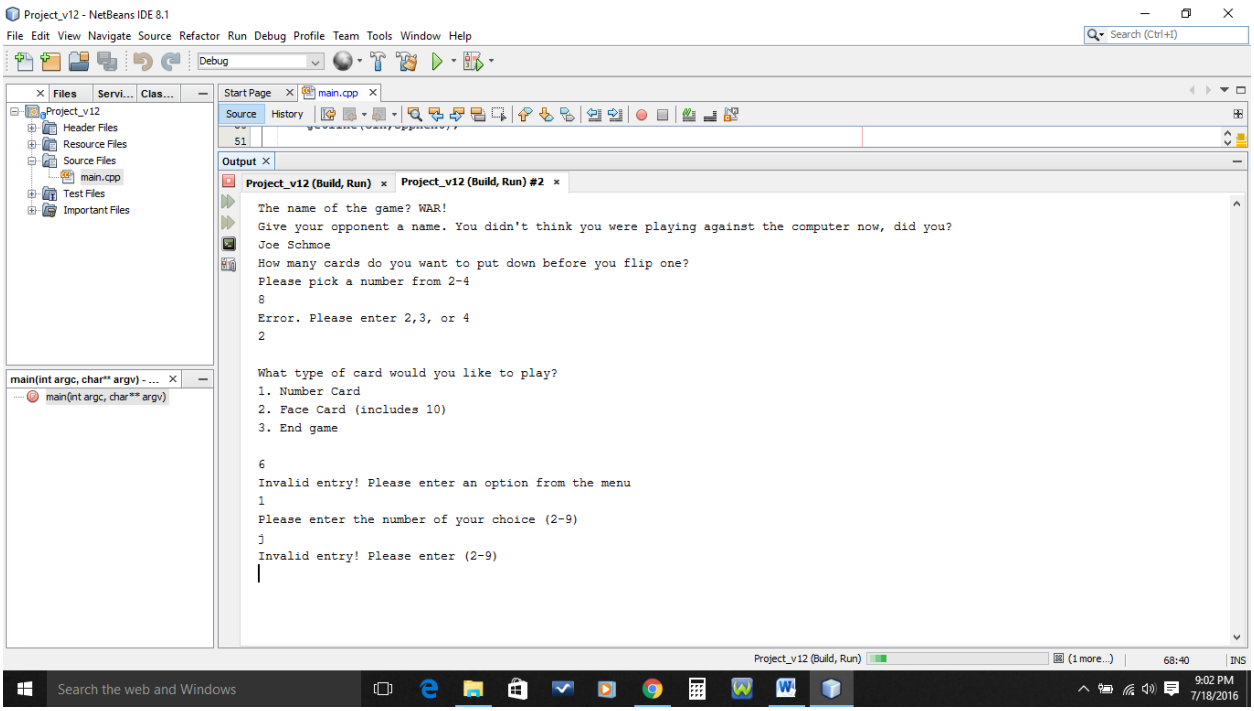
Conditional Statement	Frequency	Starting Location
if	2	Line 102,272
if/else	1	Line 297
if/else if	4	Line 77,126,189,248
switch	1	Line 103

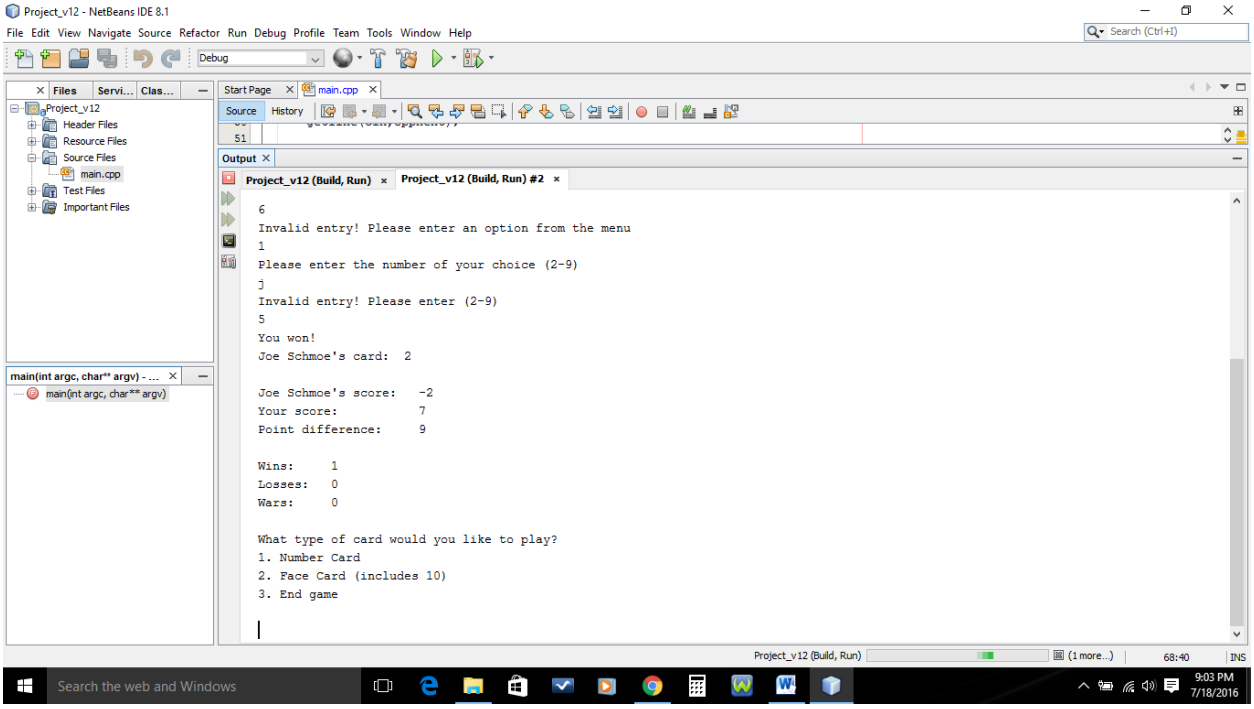
Loops:

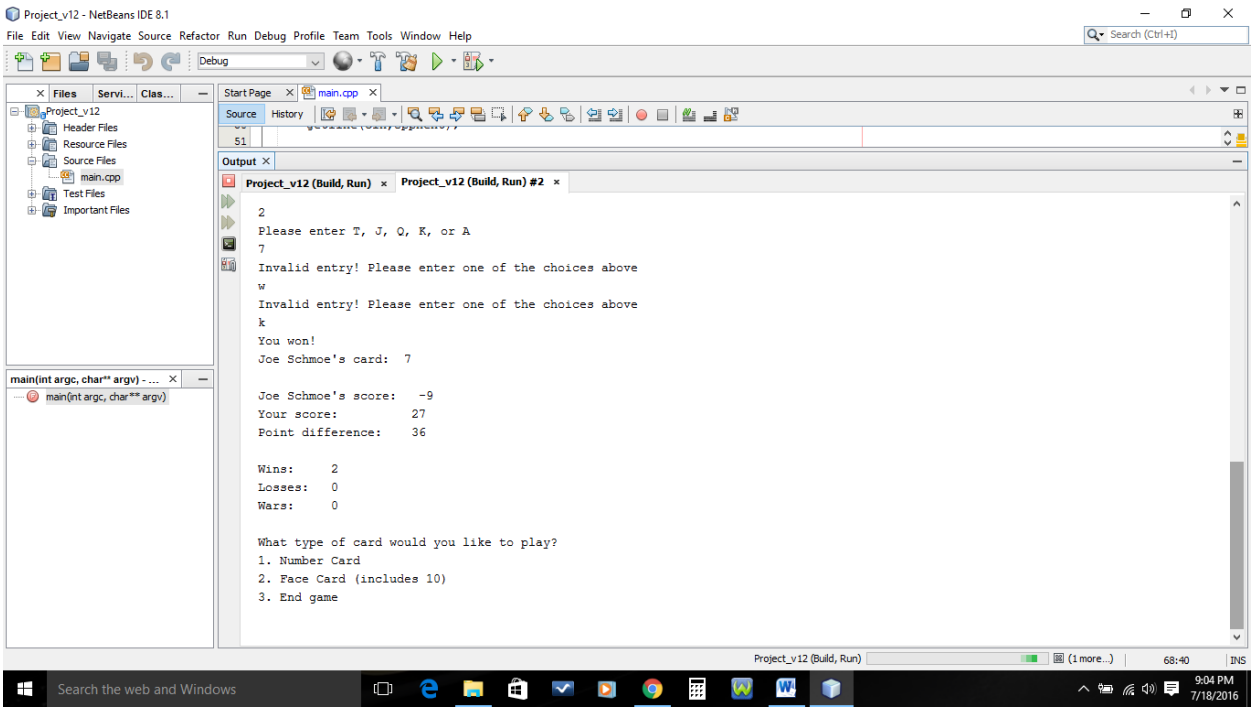
Loops	Frequency	Starting Location
for	4	Line 149,180,207,238
while	9	Line 57,72,82,92,154,170, 202,212,228
do-while	1	Line 63

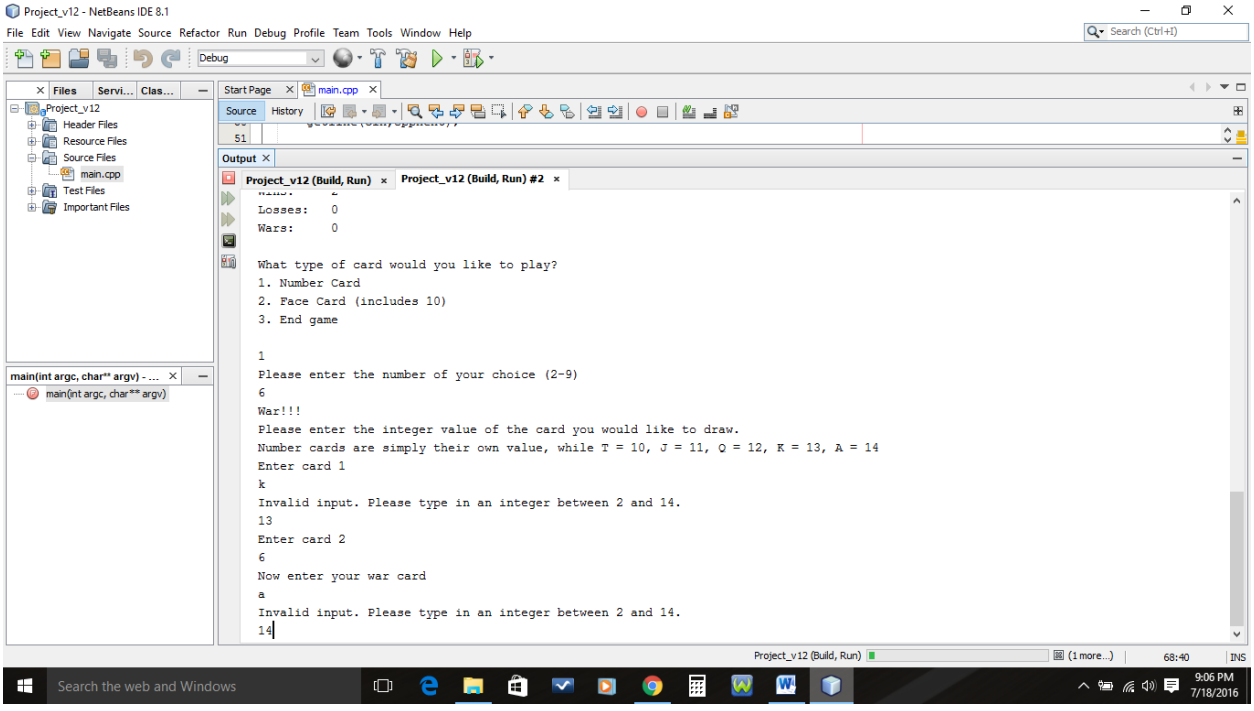
Proof of a Working Product

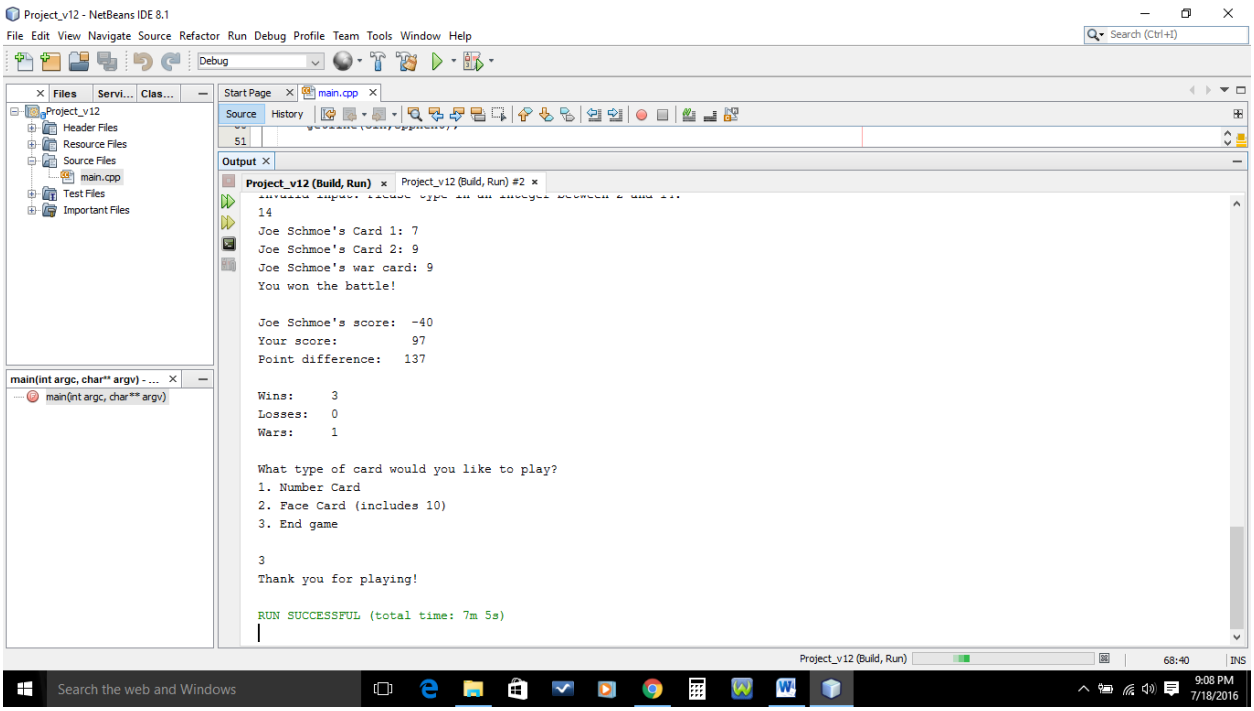
In the event, that my program does not work once it reaches Dr. Lehr, I have provided some screenshots that prove that the program did work at one time on the next few pages.

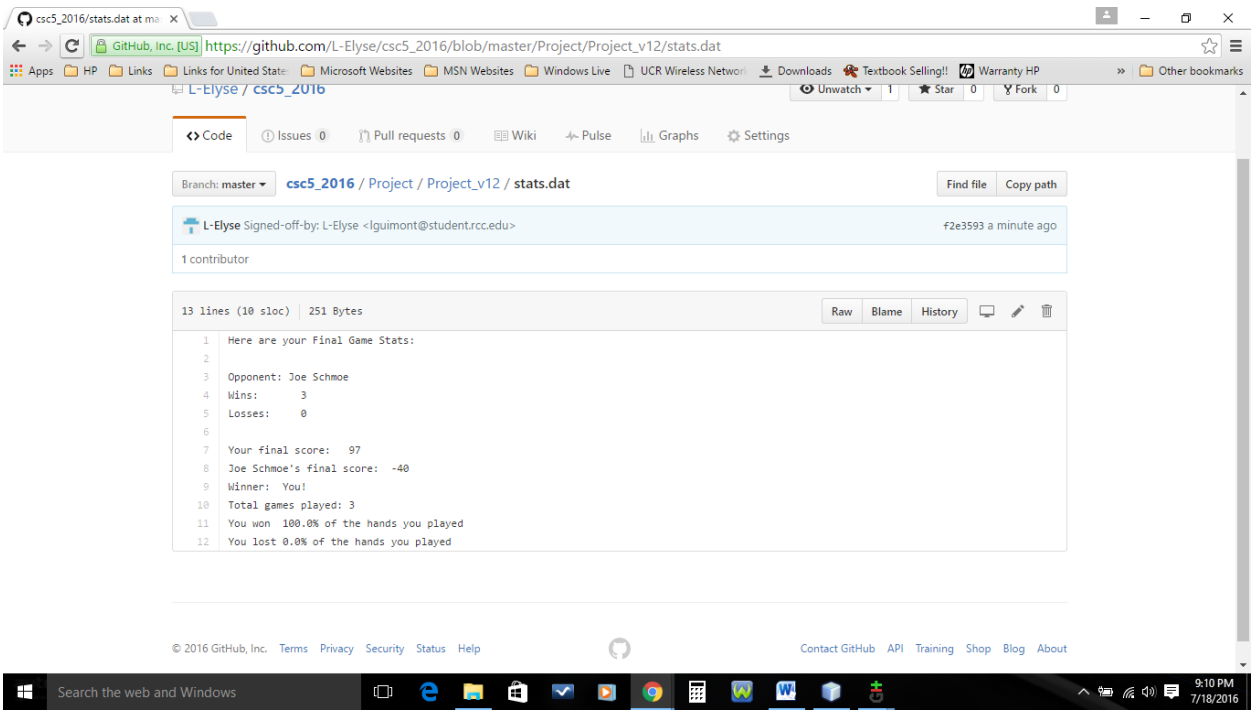

Project_v12 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Debug
Files Servi... Clas...
Project_v12
Header Files
Resource Files
Source Files
main.cpp
Test Files
Important Files
Start Page
main.cpp
Source History
Output
Project_v12 (Build, Run)
Project_v12 (Build, Run) #2
The name of the game? WAR!
Give your opponent a name. You didn't think you were playing against the computer now, did you?
Joe Schmoe
How many cards do you want to put down before you flip one?
Please pick a number from 2-4
8
Error. Please enter 2,3, or 4
2
What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game
6
Invalid entry! Please enter an option from the menu
1
Please enter the number of your choice (2-9)
j
Invalid entry! Please enter (2-9)
main(int argc, char** argv) - ...
main(int argc, char** argv)
Project_v12 (Build, Run)
(1 more...)
68:40
INS
Search the web and Windows
9:02 PM
7/18/2016


Project_v12 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Debug
Files Servi... Clas...
Project_v12
Header Files
Resource Files
Source Files
main.cpp
Test Files
Important Files
Start Page
main.cpp
Source History
Output
Project_v12 (Build, Run)
Project_v12 (Build, Run) #2
6
Invalid entry! Please enter an option from the menu
1
Please enter the number of your choice (2-9)
j
Invalid entry! Please enter (2-9)
5
You won!
Joe Schmoe's card: 2
Joe Schmoe's score: -2
Your score: 7
Point difference: 9
Wins: 1
Losses: 0
Wars: 0
What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game
main(int argc, char** argv) - ...
main(int argc, char** argv)
Project_v12 (Build, Run)
(1 more...)
68:40
INS
Search the web and Windows
9:03 PM
7/18/2016


Project_v12 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Debug
Files Servi... Clas...
Project_v12
Header Files
Resource Files
Source Files
main.cpp
Test Files
Important Files
Start Page
main.cpp
Source History
Output
Project_v12 (Build, Run)
Project_v12 (Build, Run) #2
2
Please enter T, J, Q, K, or A
7
Invalid entry! Please enter one of the choices above
w
Invalid entry! Please enter one of the choices above
k
You won!
Joe Schmoe's card: 7
Joe Schmoe's score: -9
Your score: 27
Point difference: 36
Wins: 2
Losses: 0
Wars: 0
What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game
main(int argc, char** argv) - ...
main(int argc, char ** argv)
Project_v12 (Build, Run)
(1 more...)
68:40
INS
Search the web and Windows
9:04 PM
7/18/2016


Project_v12 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
Debug
Files Servi... Clas...
Project_v12
Header Files
Resource Files
Source Files
main.cpp
Test Files
Important Files
Start Page
main.cpp
Source History
Output
Project_v12 (Build, Run)
Project_v12 (Build, Run) #2
Losses: 0
Wars: 0
What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game
1
Please enter the number of your choice (2-9)
6
War!!!
Please enter the integer value of the card you would like to draw.
Number cards are simply their own value, while T = 10, J = 11, Q = 12, K = 13, A = 14
Enter card 1
k
Invalid input. Please type in an integer between 2 and 14.
13
Enter card 2
6
Now enter your war card
a
Invalid input. Please type in an integer between 2 and 14.
14
main(int argc, char** argv) - ...
main(int argc, char ** argv)
Project_v12 (Build, Run)
(1 more...)
68:40
INS
Search the web and Windows
9:06 PM
7/18/2016


Project_v12 - NetBeans IDE 8.1
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Debug
Project_v12
Header Files
Resource Files
Source Files
main.cpp
Test Files
Important Files
Start Page
main.cpp
Source
History
Output
Project_v12 (Build, Run)
Project_v12 (Build, Run) #2
14
Joe Schmoe's Card 1: 7
Joe Schmoe's Card 2: 9
Joe Schmoe's war card: 9
You won the battle!
Joe Schmoe's score: -40
Your score: 97
Point difference: 137
Wins: 3
Losses: 0
Wars: 1
What type of card would you like to play?
1. Number Card
2. Face Card (includes 10)
3. End game
3
Thank you for playing!
RUN SUCCESSFUL (total time: 7m 5s)
main(int argc, char** argv)
Search the web and Windows
9:08 PM
7/18/2016


csc5_2016/stats.dat at ma
https://github.com/L-Elyse/csc5_2016/blob/master/Project/Project_v12/stats.dat
L-Elyse / csc5_2016
Code
Issues 0
Pull requests 0
Wiki
Pulse
Graphs
Settings
Branch: master
csc5_2016 / Project / Project_v12 / stats.dat
Find file
Copy path
L-Elyse Signed-off-by: L-Elyse <lguimont@student.rcc.edu>
f2e3593 a minute ago
1 contributor
13 lines (10 sloc) 251 Bytes
Raw
Blame
History
Here are your Final Game Stats:
Opponent: Joe Schmoe
Wins: 3
Losses: 0
Your final score: 97
Joe Schmoe's final score: -40
Winner: You!
Total games played: 3
You won 100.0% of the hands you played
You lost 0.0% of the hands you played
© 2016 GitHub, Inc. Terms Privacy Security Status Help
Contact GitHub API Training Shop Blog About
Search the web and Windows
9:10 PM
7/18/2016

References

1. Dr. Lehr's Lectures & Lab

2. "Starting Out with C++: From Control Structures through Objects" Gaddis, Tony. 8th Edition. (Textbook)

3. www.cplusplus.com (only for the use of cin.clear();)

Program

```
/*
 * File:   main.cpp
 * Author: Laurie Guimont
 * Created on July 24, 2016, 11:15 PM
 * Purpose: War Card Game Enhancement
 */

//System Libraries
#include <iostream> //Input/Output Stream Library
#include <iomanip>  //Formatting Library
#include <ctime>    //Unique Seed Value Library
#include <cstdlib>  //Random Value Library
#include <string>   //String Library
#include <fstream>  //File I/O
#include <cmath>    //Math Library
using namespace std;

//User Libraries

//Global Constants--ONLY for 2D Array
const int COL=1;

//Function Prototypes
int facdDwn(int &);
unsigned int menuOpt(unsigned int &);
int pckCard(int, char &);
int cardVal(char,int &);
void win();
void loss();
int warArry(int [],int,int &,int,int);
int sumArry(int [],int,int);
int warCard(int &,int,int);
int cwrArry(int [],int,int &,int,int,string);
int cwrCard(int &,int,int,string);
void stats(string,int,int,int,int,int);
void ldrbrd(const int [][COL],int);
void sortBrd(int [][COL],int);
void readldr(char [],int [][COL],int);
void prntldr(int [][COL],int);
```

```
//Execution Begins Here!
int main(int argc, char** argv) {
    //Set the Random Number Seed
    srand(static_cast<unsigned int>(time(0)));
    //Declare variables, no doubles
    string oppnent;      //Who you will be playing
    unsigned int choice; //User menu option
    char cchoice;        //User input representing card they want to play
    int number;          //Random number chosen set to present time
    int value;           //Value of each card
    const int MIN=2;     //Minimum value to choose from
    const int MAX=14;    //Maximum value to choose from
    int warcnt;          //Number of faced down cards before flipping in war
    int warnum,cwarnum;  //Card choice during war
    unsigned int nwins=0,nlosses=0,nwars=0,ngames;
    int score=0,wrscore=0,cscore=0,cwscore=0;
    float pwins,plosses;
    string winner;
    ofstream out;

    //Open File & Enter Primary Input Data
    out.open("stats.dat");
    cout<<"The name of the game? WAR!"<<endl;
    cout<<"Give your opponent a name. You didn't think you were ";
    cout<<"playing against the computer now, did you?"<<endl;
    getline(cin,oppnent);

    //Establish Number of "Faced Down" Cards for the Game
    facdDwn(warcnt);

    //Process and Output the Data in the Loop
    do{
        //Get Menu & Select Card
        menuOpt(choice);
        pckCard(choice,cchoice);

        //Process the card choice
        if(choice!=3){
            //Call Function & Return Value
            cardVal(cchoice,value);

            //Determine win, loss, or war
            number = (rand() % (MAX - MIN + 1)) + MIN;
            if(value>number){
                nwins+=1;
                score=score+value+number;
                cscore-=number;
                win();
                cout<<oppnent<<"'s card:  "<<number<<endl;
            }
            else if(value<number){
                nlosses+=1;
                score-=value;
                cscore=cscore+value+number;
                loss();
```

```
    cout<<oppnent<<"'s card:  "<<number<<endl;
}
else{
    nwars+=1;
    //Declare Array Variables
    const int SIZE=warcnt;
    int war[SIZE];
    int cwar[SIZE];

    //Player Process
    wrscore=warArry(war,SIZE,warnum,MIN,MAX);
    warCard(warnum,MIN,MAX);

    //Comp Process
    cwscore=cwrArry(cwar,SIZE,cwarnum,MIN,MAX,oppnent);
    cwrCard(cwarnum,MIN,MAX,oppnent);

    //Compare Cards
    if(warnum>cwarnum){
        nwins+=1;
        score=score+value+number+wrscore+cwscore+warnum+cwarnum;
        cscore=cscore-number-cwscore-cwarnum;
        cout<<"You won the battle!"<<endl;
    }
    else if (warnum<cwarnum){
        nlosses+=1;
        score=score-value-warnum-wrscore;
        cscore=cscore+value+number+cwscore+wrscore+cwarnum+warnum;
        cout<<"You lost this battle."<<endl;
    }
    else{
        while(warnum==cwarnum){  //Must War Again!
            nwars+=1;
            wrscore=warArry(war,SIZE,warnum,MIN,MAX);
            warCard(warnum,MIN,MAX);

            cwscore=cwrArry(cwar,SIZE,cwarnum,MIN,MAX,oppnent);
            cwrCard(cwarnum,MIN,MAX,oppnent);

            if(warnum>cwarnum){
                nwins+=1;
                score=score+value+number+wrscore+cwscore+warnum+
                        cwarnum;
                cscore=cscore-number-cwscore-cwarnum;
                cout<<"You won the battle!"<<endl;
            }
            else if (warnum<cwarnum){
                nlosses+=1;
                score=score-value-warnum-wrscore;
                cscore=cscore+value+number+cwscore+wrscore+
                        cwarnum+warnum;
                cout<<"You lost this battle."<<endl;
            }
        }
    }
```

```
            }
            //Game Stats
            stats(oppnent,score,cscore,nwins,nlosses,nwars);
        }
    }
    while(choice!=3);

    //End Game
    cout<<"Thank you for playing!"<<endl<<endl;

    //Show Leaderboard to Screen
    const int ROW=5;
    int board[ROW][COL];

    cout<<"Leaderboard:"<<endl;
    cout<<"Top 5 Scores"<<endl;
    readldr("leaderboard.dat",board,ROW);
    sortBrd(board,ROW);
    prntldr(board,ROW);

    //Finishing Stats - Output to a File
    out<<"Here are your Final Game Stats:"<<endl<<endl;
    out<<"Opponent: "<<oppnent<<endl;
    out<<"Wins:    "<<setw(4)<<nwins<<endl;
    out<<"Losses:  "<<setw(4)<<nlosses<<endl;
    out<<endl;
    out<<"Your final score: "<<setw(4)<<score<<endl;
    out<<oppnent<<"'s final score: "<<setw(4)<<cscore<<endl;

    //Determine Winner of Game
    if(score>cscore)
        winner="You!";
    else
        winner=oppnent;
    out<<"Winner:  "<<winner<<endl;

    //Calculate Number of Games
    ngames=nwins+nlosses;
    out<<"Total games played: "<<ngames<<endl;

    //Calculate Percentage of Wins and Losses
    pwins=static_cast<float>(nwins)/ngames;
    plosses=static_cast<float>(nlosses)/ngames;

    //Output Percentage
    out<<fixed<<setprecision(1)<<showpoint;
    out<<"You won  "<<pwins*100<<"% of the hands you played"<<endl;
    out<<"You lost "<<plosses*100<<"% of the hands you played"<<endl;

    //Close File & Exit Stage Right!
    out.close();
    return 0;
    }

void sortBrd(int a[][COL],int r){
```

```
    //Declare Variables
    bool swap;
    int temp;
    //Sort
    do{
        swap=false;
        for(int i=0;i<r-1;i++){
            for(int j=0;j<COL;j++){
                if(a[i][j]<a[i+1][j]){
                    temp=a[i][j];
                    a[i][j]=a[i+1][j];
                    a[i+1][j]=temp;
                    swap=true;
                }
            }
        }
    }
    while(swap);
}

void readldr(char fn[],int a[][COL],int r){
    //Declare the file
    ifstream in;
    //Open the file
    in.open(fn);
    //Send the array to the file
    for(int i=0;i<r;i++){
        in>>a[i][0];
    }
    //Close the file
    in.close();
}
void prntldr(int a[][COL],int r){
    for(int i=0;i<r;i++){
        cout<<setw(7)<<a[i][0]<<endl;
    }
}

void ldrbrd(const int a[][COL],int r){
    for(int i=0;i<r;i++){
        for(int j=0;j<COL;j++){
            cout<<"Player "<<i+1<<" : "<<setw(4)<<a[i][j]<<endl;
        }
    }
}

void stats(string name,int x,int y,int win,int loss,int war){
    cout<<endl;
    cout<<name<<"'s score: "<<setw(4)<<y<<endl;
    cout<<"Your score:        "<<setw(4)<<x<<endl;
    cout<<"Point difference:  "<<setw(4)<<abs(x-y)<<endl;

    if(x<0){
        cout<<"Oh no! You're in the negative!"<<endl;
        cout<<"You need to score "<<abs(x)<<" points to get out ";
```

```
        cout<<"the red zone"<<endl;
    }
    cout<<endl;
    cout<<"Wins:   "<<setw(3)<<win<<endl;
    cout<<"Losses: "<<setw(3)<<loss<<endl;
    cout<<"Wars:   "<<setw(3)<<war<<endl;
}

int cwrCard(int &number,int min,int max,string name){
    //Opponent's War Card
    number = (rand() % (max - min + 1)) + min;
    cout<<name<<"'s war card: "<<number<<endl;
}

int warCard(int &number,int min, int max){
    //Player's War Card
    cout<<"Now enter your war card"<<endl;
    cin>>number;

    //Input Validation
    while(!(number)||number<min||number>max){
        cin.clear();
        cin.ignore();
        cout<<"Invalid input. Please type in an integer";
        cout<<" between 2 and 14."<<endl;
        cin>>number;
    }
    return number;
}

int sumArry(int a[],int n){
    int sum=0;
    for(int i=0;i<n;i++){
        sum+=a[i];
    }
    return sum;
}

int cwrArry(int a[],int n,int &val,int min,int max,string name){
    //Opponent's "Faced Down" Cards
    cout<<name<<"'s 'Faced Down' Cards: ";
    for(int i=0;i<n;i++){
        a[i] = (rand() % (max - min + 1)) + min;
        cout<<a[i]<<" ";
    }
    cout<<endl;
    val=sumArry(a,n);
    return val;
}

int warArry(int a[],int n,int &val,int min, int max){
    cout<<"War!!!"<<endl;
    cout<<"Please enter the integer value of the card you ";
    cout<<"would like to draw."<<endl;
    cout<<"Number cards are simply their own value, while T = 10,"
```

```
            " J = 11, Q = 12, K = 13, A = 14"<<endl;

    //Player's "Faced Down" Cards
    for(int i=0;i<n;i++){
        cout<<"Enter card "<<i+1<<endl;
        cin>>a[i];

        //Input Validation
        while(!(a[i])||a[i]<min||a[i]>max){
            cin.clear();
            cin.ignore();
            cout<<"Invalid input. Please type in an integer";
            cout<<" between 2 and 14."<<endl;
            cin>>a[i];
        }
        //Add Elements in Array
        val=sumArry(a,n);

    }return val;
}

void loss(){
    cout<<"Sorry. You lost."<<endl;
    return;
}

void win(){
    cout<<"You won!"<<endl;
    return;
}

int cardVal(char choice,int &number){
    switch(choice){
        case 'a':
        case 'A':number=14;break;
        case 'k':
        case 'K':number=13;break;
        case 'q':
        case 'Q':number=12;break;
        case 'j':
        case 'J':number=11;break;
        case 't':
        case 'T':number=10;break;
        case '9':
        case '8':
        case '7':
        case '6':
        case '5':
        case '4':
        case '3':
        case '2':number=(choice-48);break;
    }
    return number;
}
```

```
int pckCard(int menu, char &card){
    if(menu==1){
        cout<<"Please enter the number of your choice (2-9)"<<endl;
        cin>>card;

        //Input Validation
        while(card<'2'||card>'9'){
            cout<<"Invalid entry! Please enter (2-9)"<<endl;
            cin>>card;
        }
    }
    else if(menu==2){
        cout<<"Please enter T, J, Q, K, or A"<<endl;
        cin>>card;

        //Input Validation
        while(card!='a'&&card!='A'&&card!='k'&&card!='K'&&
                card!='q'&&card!='Q'&&card!='j'&&card!='J'&&
                card!='t'&&card!='T'){
            cout<<"Invalid entry! Please enter one of the choices "
                    "above"<<endl;
            cin>>card;
        }
    }
    return card;
}

unsigned int menuOpt(unsigned int &option){
    cout<<endl;
    cout<<"What type of card would you like to play?"<<endl;
    cout<<"1. Number Card"<<endl;
    cout<<"2. Face Card (includes 10)"<<endl;
    cout<<"3. End game"<<endl<<endl;
    cin>>option;

    //Input Validation
    while(option<1||option>3){
        cout<<"Invalid entry! Please enter an option from the menu"<<endl;
        cin>>option;
    }
    return option;
}

int facdDwn(int &number){
    cout<<"How many cards do you want to put down before you flip one?"<<endl;
    cout<<"Please pick a number from 2-4"<<endl;
    cin>>number;

    //Input Validation
    while(number<2||number>4){
        cout<<"Error. Please enter 2,3, or 4"<<endl;
        cin>>number;
    }
    return number;
}
```