**Project 1**

Title

**Card War Game**

Course

**CIS-5**

Section

**40570**

Due Date

**February 3, 2022**

Author

**Liliana Darch**

**Introduction**

**Title:**

Card War Game

**Number of players:**

2

**How to play:**

The game uses a standard 52-card deck of playing cards divided evenly and randomly between two players. Each player gets 26 random cards face down. Both player flip the top card at the same time, compare both cards and the player with the higher card wins both cards.

A war consist in a tie between the two player because both players fliped identical cards, each player lays three cards face-down, then each player flips one card face-up, compare the new face-up cards played and the player with the higher rank wins all the cards.

If the cards tie again then you have another war and repear the process of the war and continue the process until there is a winner. If one player doesn’t have more cards to lay face-down during the war, he or she will lose the war.

The first player to collect all the cards wins.

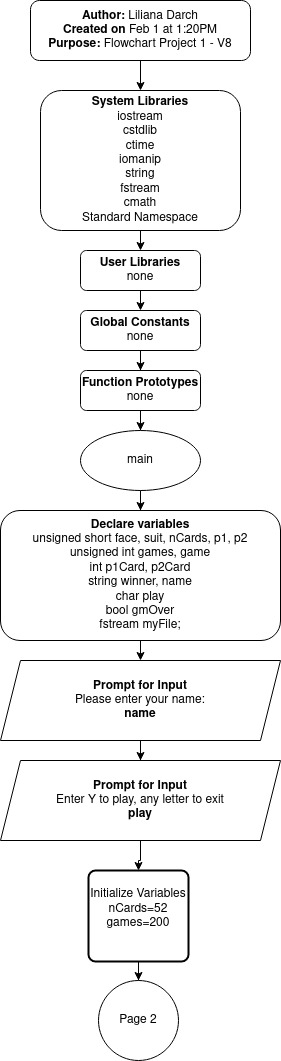
**Card denomination:**

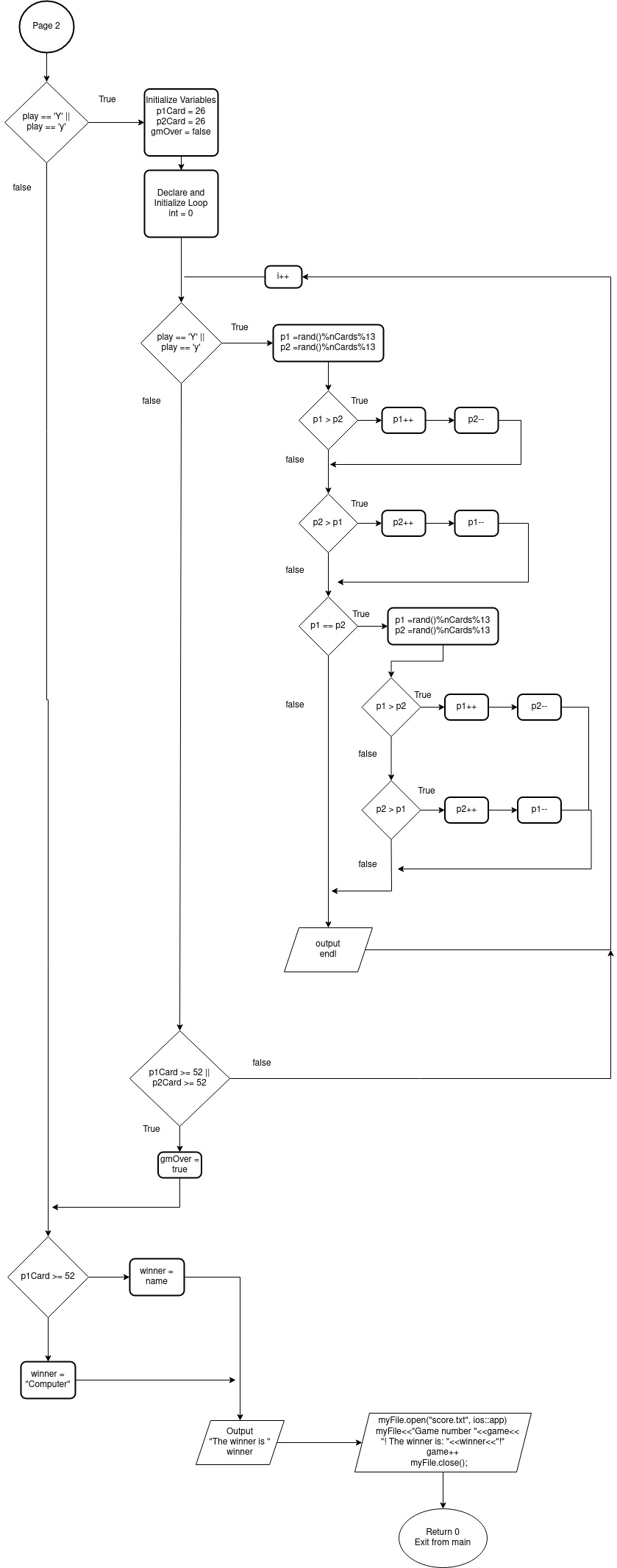
A is the highest card = 13

2 is the lowest card = 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Card | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | J | Q | K | A |
| Equal to | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

**Flowchart**





**Pseudocode**

DECLARE a variable face to hold the face of the cards

DECLARE a variable suit to hold the suit of the cards

DECLARE a variable nCards for number of cards of the deck

DECLARE a variable p1 for player one

DECLARE a variable p2 for player two

DECLARE a variable games to count the rounds of one game

DECLARE a variable game to count the score of each game in the file

DECLARE a variable p1Cards to count the total of cards player one has each round

DECLARE a variable p2Cards to count the total of cards player two has each round

DECLARE a variable winner to hold the winner of the game

DECLARE a variable name to hold the name of player one

DECLARE a variable play to hold if player one would like to play again

DECLARE a variable bool gmOver to hold if we have a winner

DECLARE a variable myFile to write on the file

LOOP: While the player wants to keep playing

GET: the users input

LOOP: Do:

IF the player1 card is higher than player2

ADD one to player1

SUBSTRACT one to player2

END IF

IF the player2 card is higher than player1

ADD one to player2

SUBSTRACT one to player1

END IF

IF player1 has the same card as player2

IF the player1 card is higher than player2

ADD four to player1

SUBSTRACT four to player2

ELSE the player2 card is higher than player1

ADD four to player2

SUBSTRACT four to player1

END IF

ELSE endl

IF player1 has 52 cards or player2 has 52 cards

gmOver is equal to true

END IF

WHILE gmOver is false

IF player1 has 52 cards, assigned player 1 to winner OTHERWISE assigned “Computer”

END LOOP

**Spreadsheet – Cross Reference for Project 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Chapter** | **Section** | **Topic** | **Where Line #''s** | **Pts** | **Notes** |
| 2 | 2 | cout | 54 |  |  |
|  | 3 | libraries | 15 to 21 | 8 | iostream, iomanip, cmath, cstdlib, fstream, string, ctime |
|  | 4 | variables/literals | 40 to 49 |  | No variables in global area, failed project! |
|  | 5 | Identifiers |  |  |  |
|  | 6 | Integers | 43 | 3 |  |
|  | 7 | Characters | 47 | 3 |  |
|  | 8 | Strings | 45 | 3 |  |
|  | 9 | Floats No Doubles | Version 2 – 40 | 3 | Using doubles will fail the project, floats OK! |
|  | 10 | Bools | 48 | 4 |  |
|  | 11 | Sizeof \*\*\*\*\* |  |  |  |
|  | 12 | Variables 7 characters or less |  |  | All variables <= 7 characters |
|  | 13 | Scope \*\*\*\*\* No Global Variables |  |  |  |
|  | 14 | Arithmetic operators |  |  |  |
|  | 15 | Comments 20%+ | 54 | 5 | Model as pseudo code |
|  | 16 | Named Constants |  |  | All Local, only Conversions/Physics/Math in Global area |
|  | 17 | Programming Style \*\*\*\*\* Emulate |  |  | Emulate style in book/in class repositiory |
|  |  |  |  |  |  |
| 3 | 1 | cin | 55 |  |  |
|  | 2 | Math Expression |  |  |  |
|  | 3 | Mixing data types \*\*\*\* |  |  |  |
|  | 4 | Overflow/Underflow \*\*\*\* |  |  |  |
|  | 5 | Type Casting | 37 | 4 |  |
|  | 6 | Multiple assignment \*\*\*\*\* |  |  |  |
|  | 7 | Formatting output | 79 | 4 |  |
|  | 8 | Strings | 152 | 3 |  |
|  | 9 | Math Library | Declare 21 , use in 86 | 4 | All libraries included have to be used |
|  | 10 | Hand tracing \*\*\*\*\*\* |  |  |  |
|  |  |  |  |  |  |
| 4 | 1 | Relational Operators |  |  |  |
|  | 2 | if | 77 | 4 | Independent if |
|  | 4 | If-else | 139 | 4 |  |
|  | 5 | Nesting | 144 | 4 |  |
|  | 6 | If-else-if | 118 | 4 |  |
|  | 7 | Flags \*\*\*\*\* |  |  |  |
|  | 8 | Logical operators | 73 | 4 |  |
|  | 11 | Validating user input | 64 | 4 |  |
|  | 13 | Conditional Operator | 89 | 4 |  |
|  | 14 | Switch | Version1 – 44 | 4 |  |
|  |  |  |  |  |  |
| 5 | 1 | Increment/Decrement | 83, 84 | 4 |  |
|  | 2 | While | 64 | 4 |  |
|  | 5 | Do-while | 71 | 4 |  |
|  | 6 | For loop | 73 | 4 |  |
|  | 11 | Files input/output both | 159 | 8 |  |
|  | 12 | No breaks in loops \*\*\*\*\*\* |  |  | Failed Project if included |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| \*\*\*\*\*\* Not required to show |  |  | Total | 100 |  |

**Code**

/\*

\* File: main.cpp

\* Author: Liliana Darch

\* Created on January 31, 2022, 11:00 PM

\* Purpose: Project Card War Game Version 8

\*/

//System Level Libraries

#include <iostream> //I/O Library

#include <cstdlib> //Random Function Library

#include <ctime> //Time Library

#include <iomanip> //Formatting Library

#include <string> //String library

#include <fstream> //File library

#include <cmath> //Math libray

using namespace std;

//User Defined Libraries

//Global constants, not Global variables

//These are recognized constants from the sciences

//Physics/Chemistry/Engineering and Conversions between

//systems of units!

//Function Prototypes

//Execution begins here!

//Execution Begins Here

int main(int argc, char\*\* argv) {

//Set Random Number seed

srand(static\_cast<unsigned int>(time(0)));

//Declare Variable Data Types and Constants

unsigned short face, suit, //variables for the face and suit of the deck of cards

nCards, //number of cards in a deck

p1, p2; //Player one and player two

unsigned int games, game; //The amount of rounds available to play

int p1Card, p2Card; //The amount of cards that each player has every round

string winner; //Holds the winner

string name; //Holds the name of player 1

char play; //Validates the user input keep playing

bool gmOver; //To check if we have a winner and a looser

fstream myFile; //File to hold the winner from each game

//Initialize Variables

cout<<"Please enter your name: "<<endl; //Ask for the player's name

cin>>name; //Read the player name

cout<<"Hi "<<name<<" are you ready to play against the computer?"<<endl;

cout<<"Enter Y to play, any letter to exit"<<endl;

cin>>play; //Read the user input about playing again

nCards=52; // The 52 cards in a deck

games=200; // Play game no more than 200 rounds

while (play == 'Y' || play == 'y'){ //Checks if the user would like to play the game

p1Card = 26; //Initialize player 1 with 26 cards

p2Card = 26; //Initialize player 2 with 26 cards

gmOver = false; //Check if there was a winner and a looser

//Start the game

do {

for(int i = 0; i<games && !gmOver; i++){

p1 = rand()%nCards%13; //Assign the random card to Player 1

p2 = rand()%nCards%13; //Assign the random card to Player 2

if (p1 > p2){ //If player 1 has the highest card

cout<<endl;

cout<<"\*\*\* ROUND NUMBER "<<setw(3)<<i<<" \*\*\*"<<endl; //Show the round number

cout<<name<<" card is "<<p1<<endl;

cout<<"Computer card is "<<p2<<endl;

cout<<name<<" has the highest card "<<p1<<endl;

p1Card++; //Adds 1 card to player 1

p2Card--; //Subtract 1 card to player 2

cout<<"-- The statistics --"<<endl;

cout<<name<<" has "<<abs(p1Card)<<" cards"<<endl;

cout<<"Computer has "<<abs(p2Card)<<" cards"<<endl;

}

if (p2 > p1){ // If player 1 has the highest card

cout<<endl;

cout<<"\*\*\* ROUND NUMBER "<<setw(3)<<i<<" \*\*\*"<<endl; //Show the round number

cout<<name<<" card is "<<p1<<endl;

cout<<"Computer card is "<<p2<<endl;

cout<<"Computer has the highest card "<<p2<<endl;

p2Card++; //Adds 1 card to player 2

p1Card--; //Subtract 1 card to player 2

cout<<"-- The statistics --"<<endl;

cout<<name<<" has "<<abs(p1Card)<<" cards"<<endl;

cout<<"Computer has "<<abs(p2Card)<<" cards"<<endl;

}

if(p1 == p2) { //If we get a match, we have a war!!

cout<<endl;

cout<<"\*\*\* ROUND NUMBER "<<setw(3)<<i<<" \*\*\*"<<endl; //Show the round number

cout<<name<<" card is "<<p1<<endl;

cout<<"Computer card is "<<p2<<endl;

cout<<"\*\*\* WE HAVE A WAR \*\*\*"<<endl;

p1 = rand()%nCards%13; //Assign a new the random card to Player 1 to play the war

p2 = rand()%nCards%13; //Assign a new the random card to Player 2 to play the war

cout<<endl;

cout<<"\*\*\*Round number "<<i<<endl; //Show the round number

if (p1 > p2){ // If player 1 has the highest card

cout<<name<<" new card is "<<p1<<endl;

cout<<"Computer new card is "<<p2<<endl;

cout<<name<<" wins the war with the card number "<<p1<<endl;

if ((p1Card <48) || (p2Card <48)) { //control the amount of cards played

p1Card = p1Card + 4; //Add 4 card to player 1

p2Card = p2Card - 4; //Subtract 4 card to player 2

cout<<"-- The statistics --"<<endl;

cout<<name<<" has "<<abs(p1Card)<<" cards"<<endl;

cout<<"Computer has "<<abs(p2Card)<<" cards"<<endl;

}

} else{ //If player 2 has the highest card

cout<<name<<" new card is "<<p1<<endl;

cout<<"Computer new card is "<<p2<<endl;

cout<<"Computer wins the war with the card number "<<p2<<endl;

if ((p1Card <=48) || (p2Card <=48)){

p2Card = p2Card + 4; //Add 4 card to player 2

p1Card = p1Card - 4; //Subtract 4 card to player 1

cout<<"-- The statistics --"<<endl;

cout<<name<<" has "<<abs(p1Card)<<" cards"<<endl;

cout<<"Computer has "<<abs(p2Card)<<" cards"<<endl;

}

}

}else {

cout<<endl;

}

if( p1Card >= 52 || p2Card >= 52 ) //Check if we have a winner every round

gmOver = true; //Set the boolean to true so the game is over

}

} while (gmOver == false); //Keep playing while bool is equal to false

winner = p1Card >= 52 ? name:"Computer"; //Assign the winner player to the winner variable

cout<<endl<<"THE WINNER IS"<<endl;

cout<<setw(10)<<winner<<endl; //Output the winner

cout<<endl<<endl;

myFile.open("score.txt", ios::app); // Open a file

myFile<<"Game number "<<game<<"! The winner is: "<<winner<<"!"<<endl; //Output in the file

game++; // counts how many games were played

myFile.close(); //close the file

cout<<"Do you want to play again?"<<endl; //Ask if the user would like to play again

cout<<"Enter Y to play, any letter to exit"<<endl;

cin>>play; //Read user input about playing again

}

//Exit stage right!

return 0;

}

**Summary**

Project size: about 172 lines

The number of variables: 13

This project includes many concepts that we learned from the chapters 1 to 5 from the book Gaddis 9th Edition. Also, it has many possibilities to be extended for next project. For example, the use functions to clean the main, the use of arrays to hold de card suits and faces.

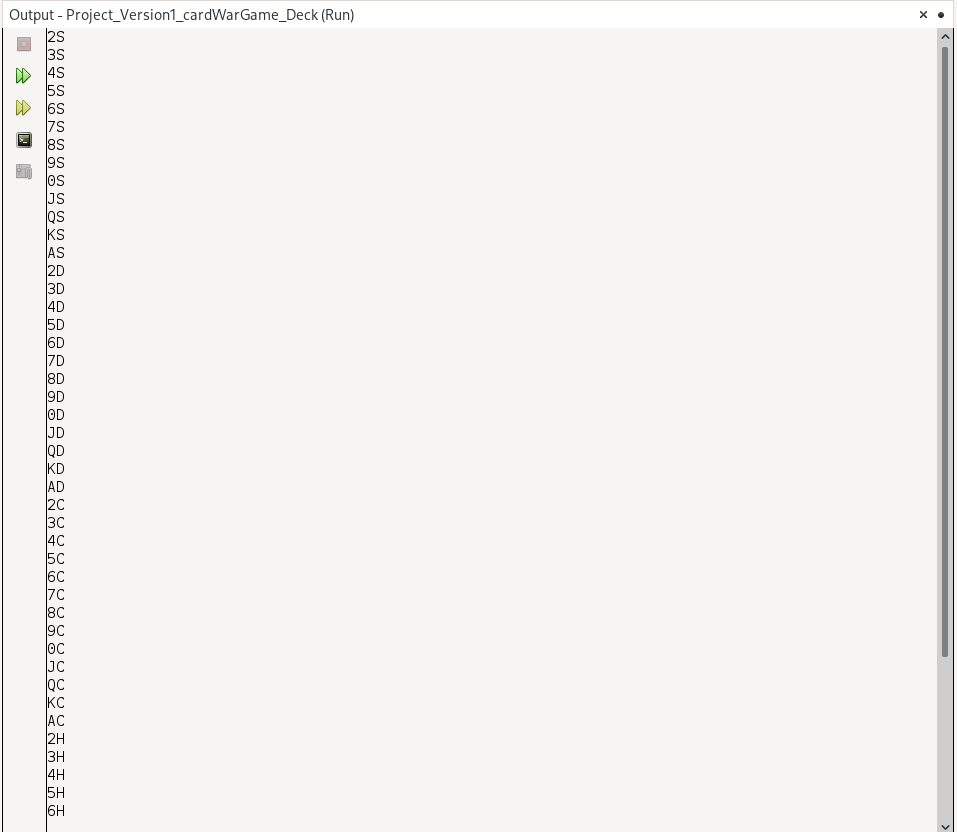
**Version 0:**

Contains the minimun and maximun numbers available using the random variable.



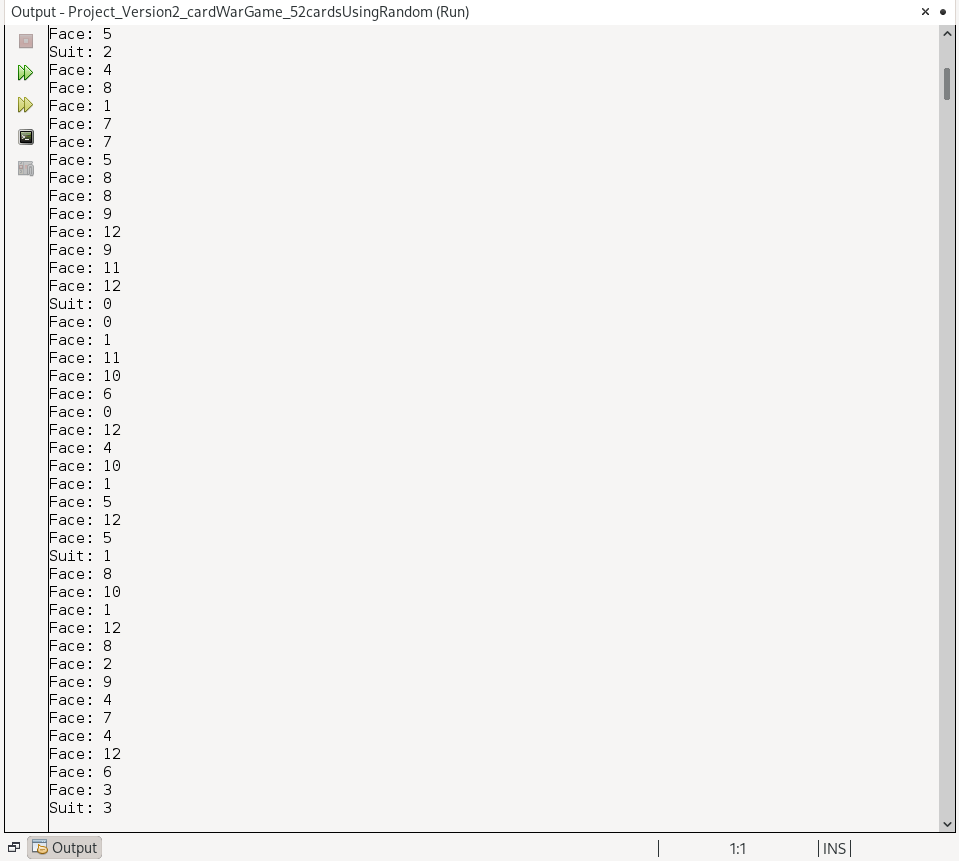
**Version 1:**

Create a file with the deck of cards. The file contains the suit H for Hearts, D for Diamonds, C for Cloves and S for Spaces. It also contains the card’s faces, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K and, A.



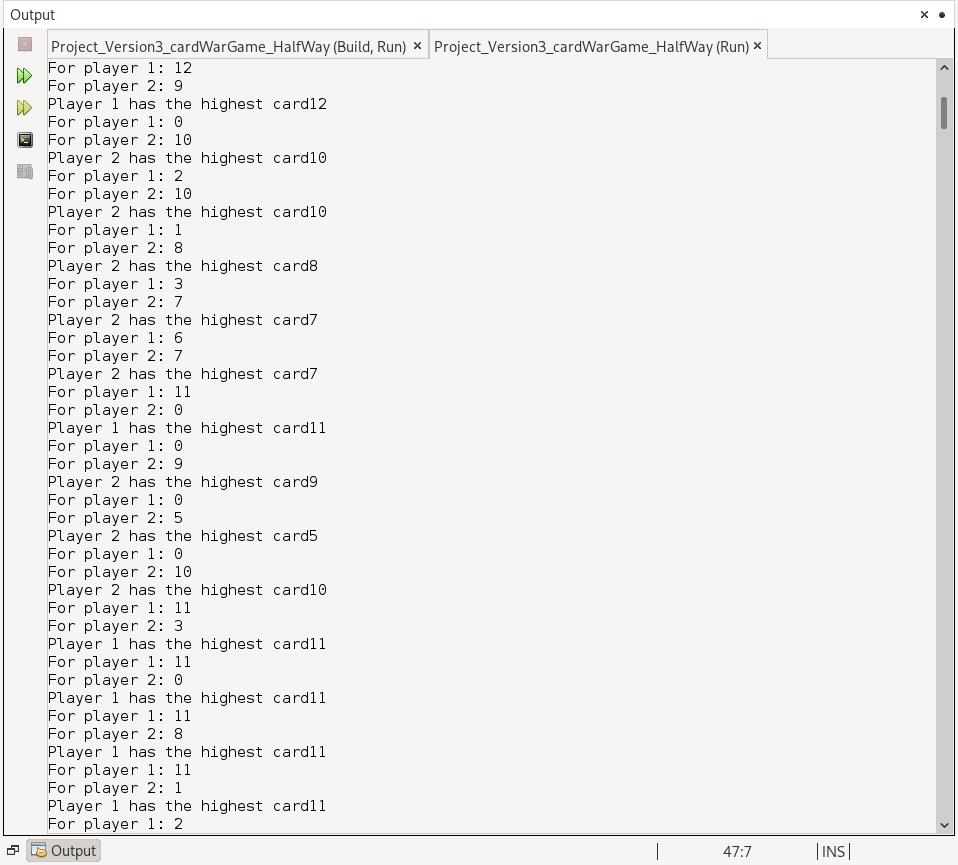
**Version 2:**

I’m using the random variable to be able to get suits from 0 to 3 that represent the hearts, diamonds, cloves and spaces. And also, I try the random variable to get the random faces represented in numbers from 0 to 12.



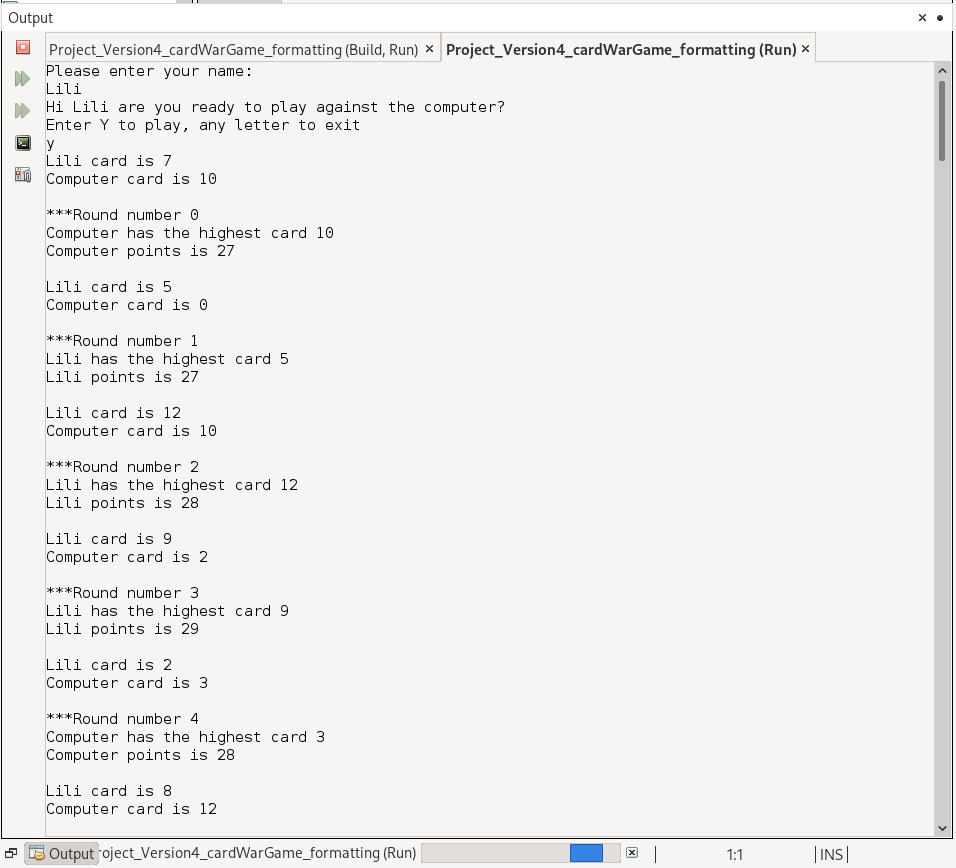
**Version 3:**

Checking which player has the higher card using if statments and conditional operator

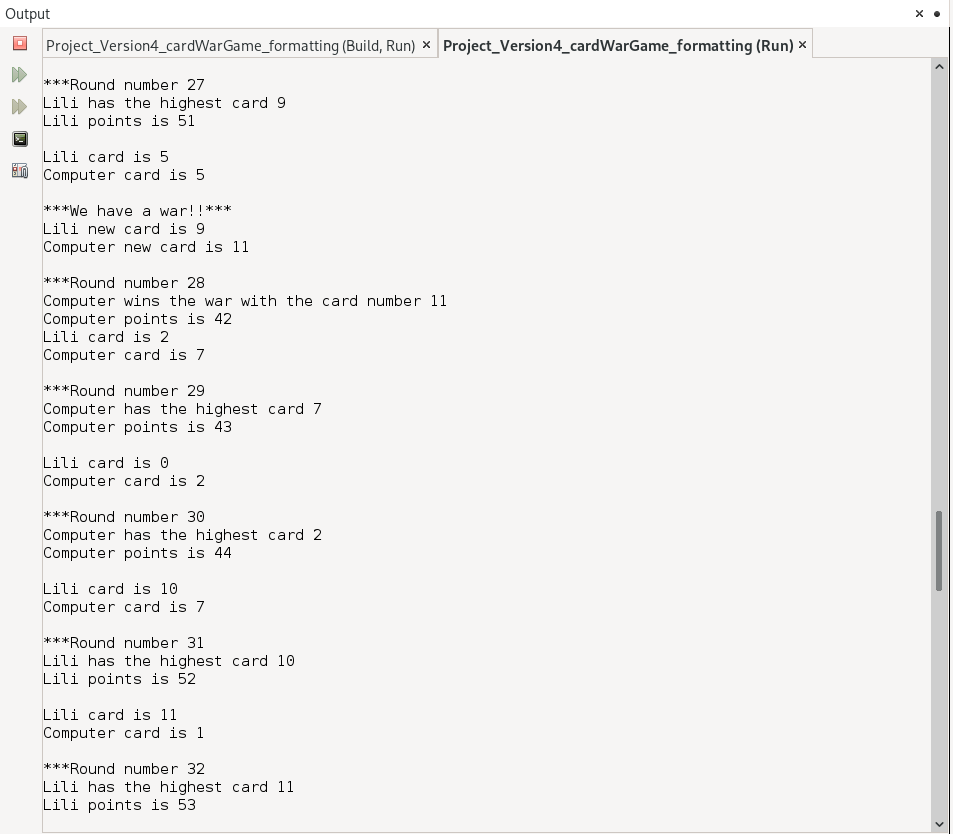


**Version 4:**

Use the string to hold the player’s name. Incorporate a do while loop to play while the player press the key ‘Y’ or ‘y’ .

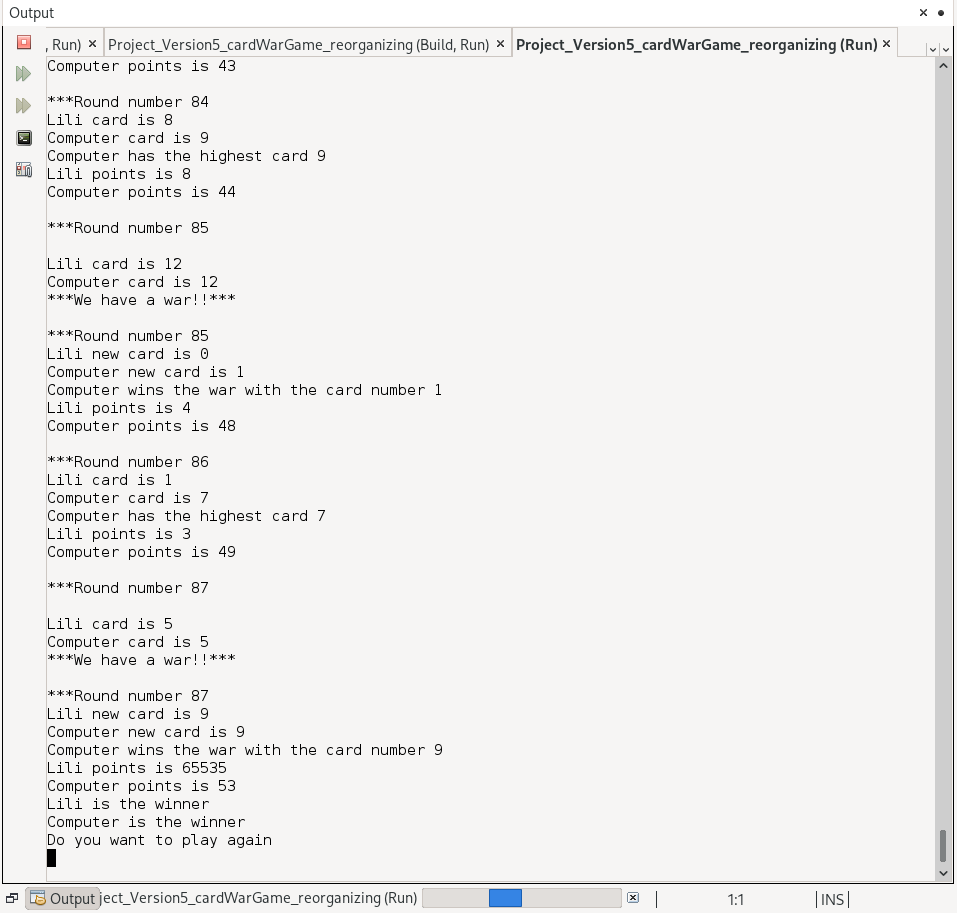


**Continue: Version 4** also check if the player 1 and player 2 have identical cards so they can play a WAR.



**Version 5:**

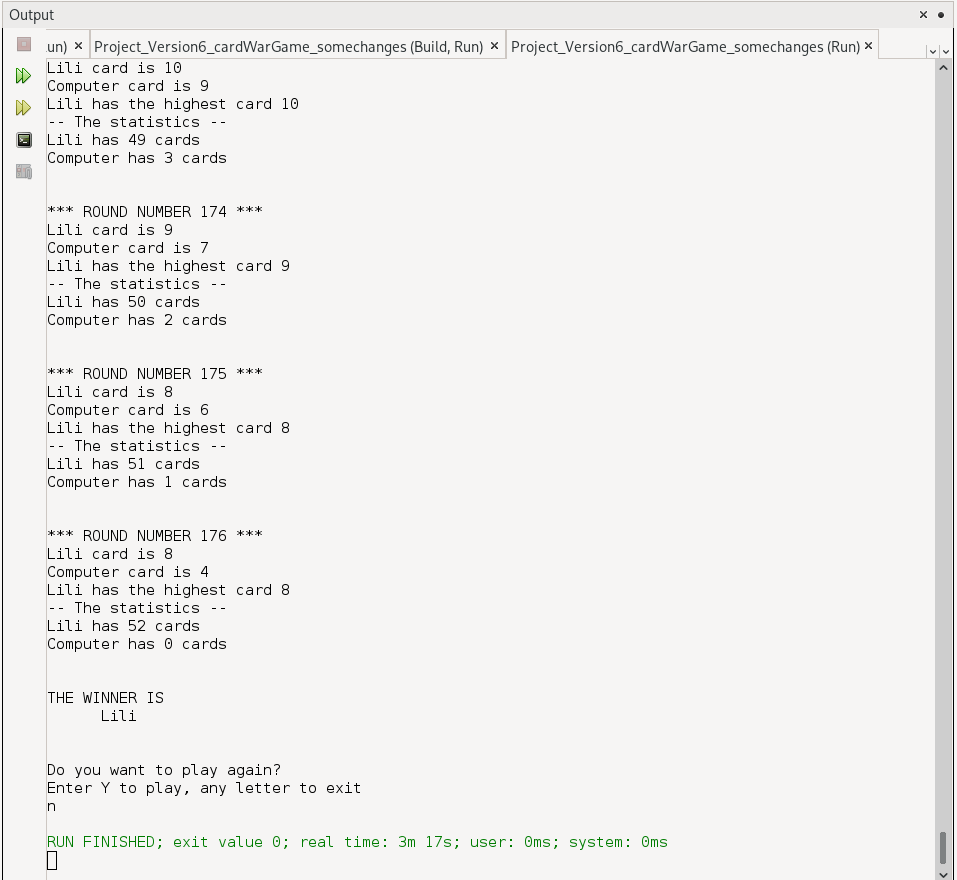
I fixed some typos but I realized that I was having an error when printing the points for the winner because I was using an unsigned short if the player have -1 card the output will show as if the player had 65535 cards instead of a negative number.



**Version 6:**

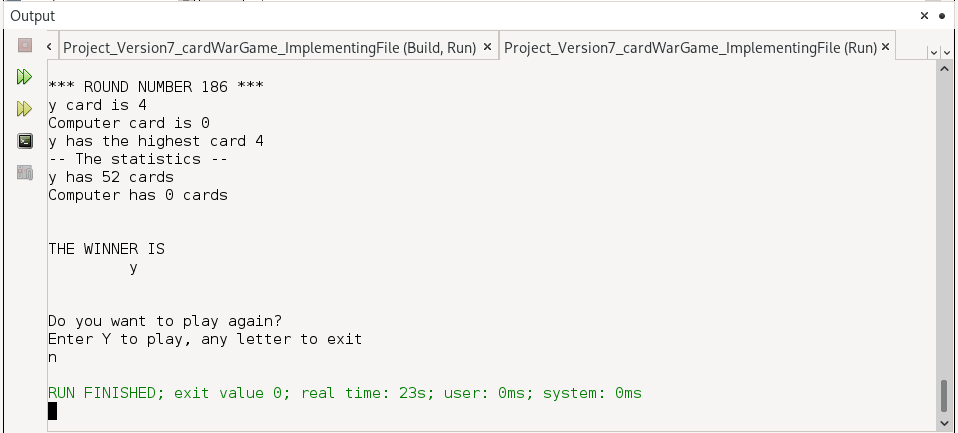
Adding a bool variable to fix the error of printing the winner pointing.

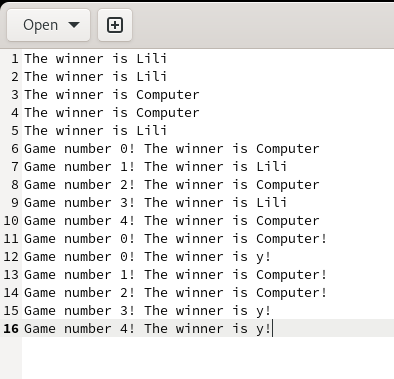
Also, spent some time working in the formatting the output to make the output console more readible.

****

**Version 7:**

Created a file document to be able to store the winner for each game played.

****

****

**Version 8:**

Is the final version of my project 1.

Fixed some formating output.

