**Project 2:**

**Blackjack 21**

**CIS-5-40651**

**Antonio Gines**

**02/08/18**

**Intro**

Most of my family has pretty consistent luck when it comes to card games(with me being the exception) and one of the most popular games amongst my family members is Blackjack 21. In a recent venture to a casino for my grandfather’s 86th birthday my aunt and uncle took me to one of the single deck poker tables and taught me the rules and when to recognize a decent hand. From their tutelage I was able to turn $60 into $120. This capital gain and the satisfaction from playing Blackjack 21 inspired me to base my project off of it.

**Rules of the Game**

Game: Blackjack 21

Blackjack is a gambling card game in which the player starts by placing a bet valued between the minimum and maximum amount of money that the house allows. The player as well as the dealer are dealt two cards and of the two cards the dealer receives, one of them is revealed. The player, with the hand they are dealt can choose to stay (keep their current cards), hit (receive another card), double (add an additional bet equal to the first one and receive a singular extra card), or, if the first two cards are equal, split (add an additional bet and split the hand into two hands where both receive an additional card). If the ultimate value of the player’s hand is 21, greater than the dealer’s hand, or if the dealer busts, the player wins and receives double his/her gross bet. If the player and the dealer have the same hand without it being a bust it is a tie and there is no gain nor loss. If the player busts or if the dealer has a higher hand, the player loses the bet.

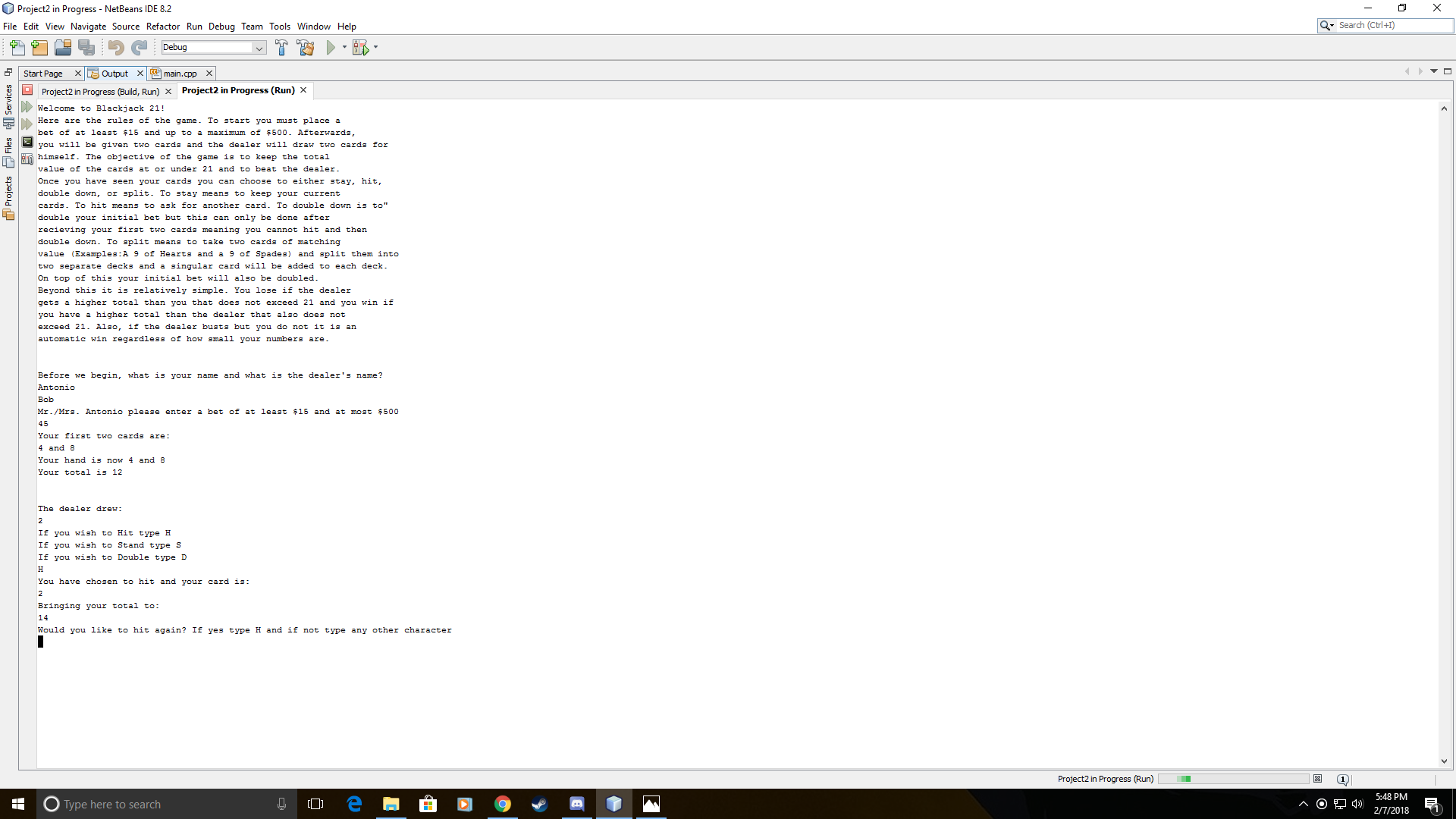
**Summary of Process**

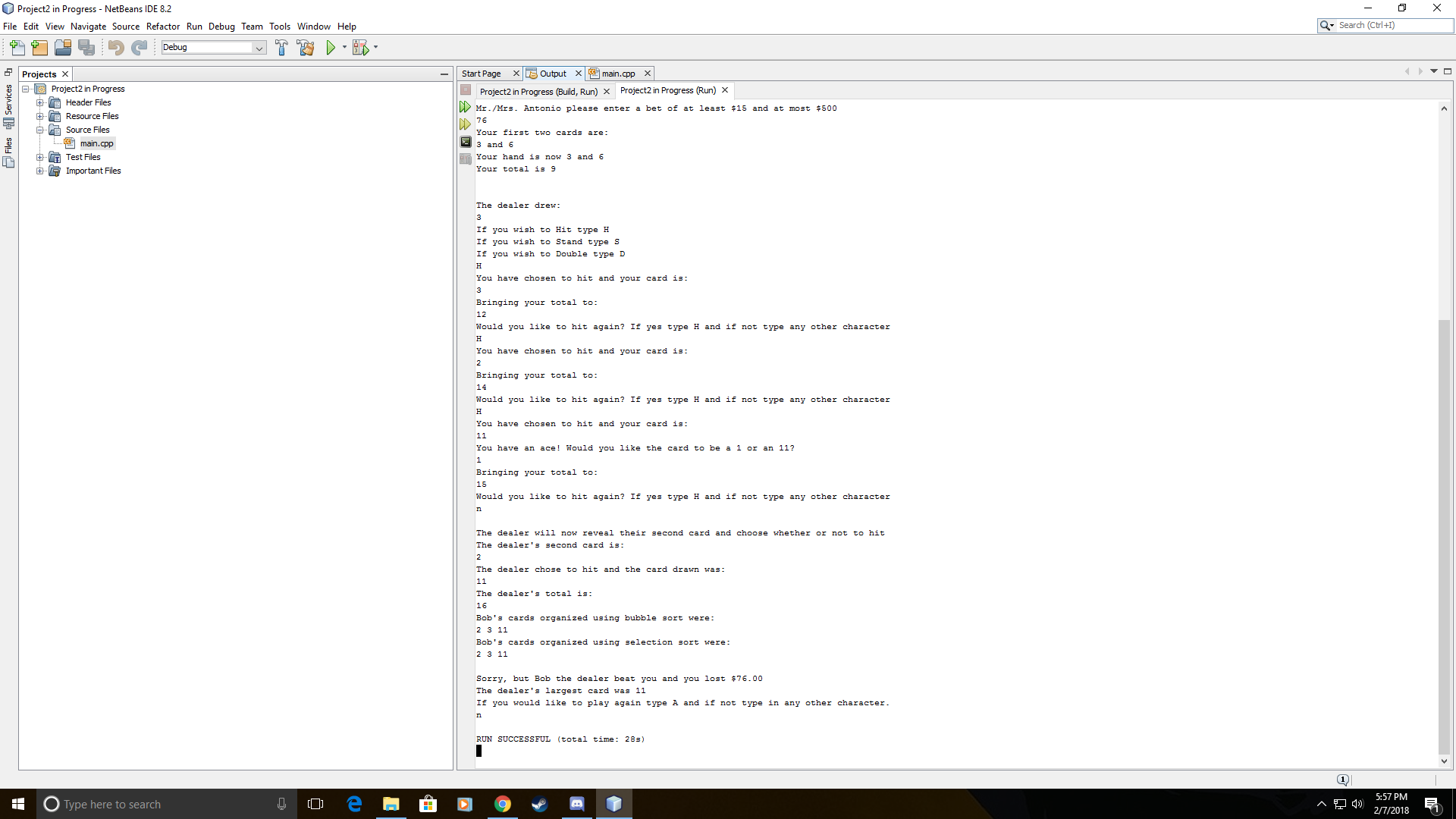
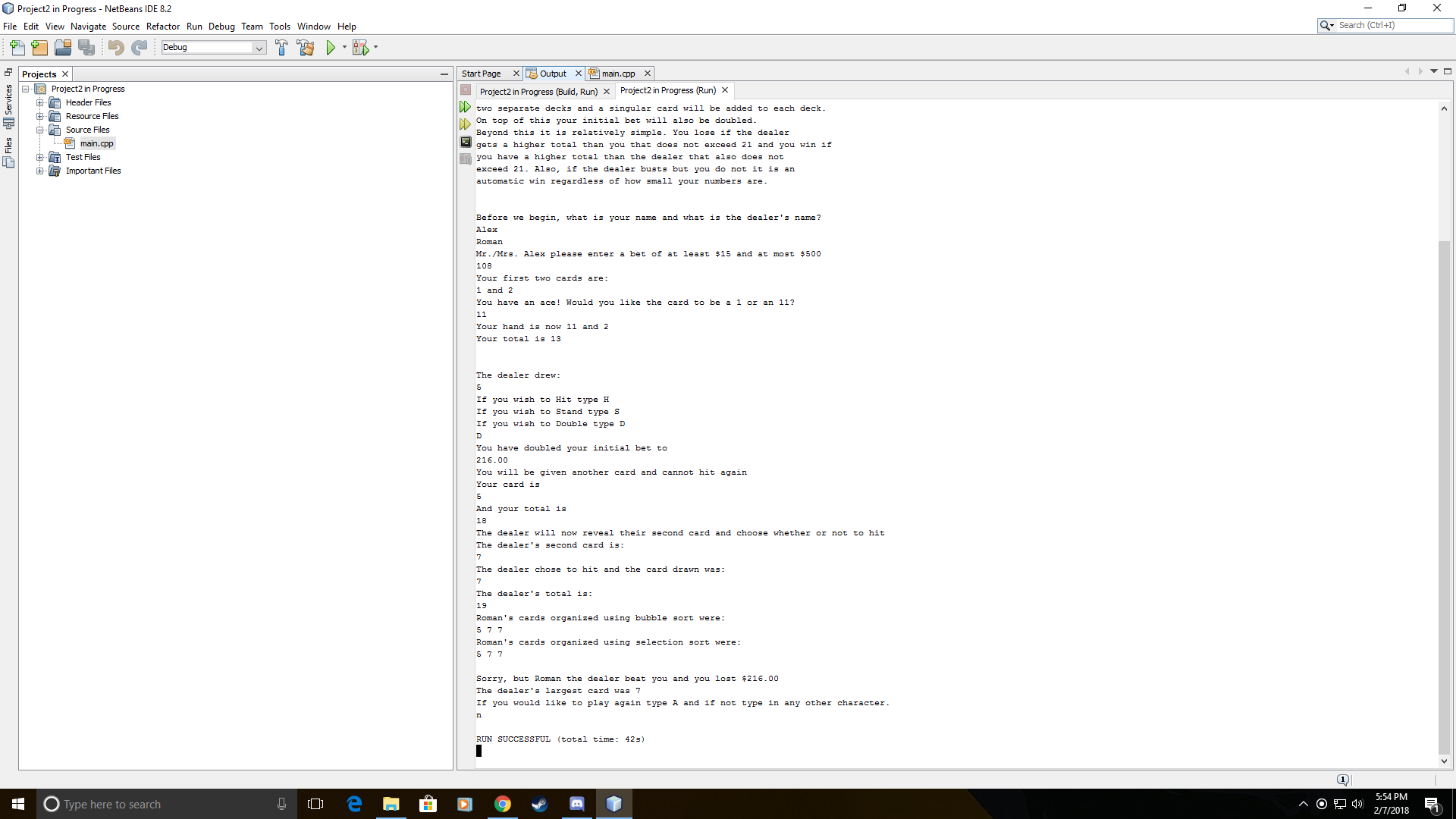
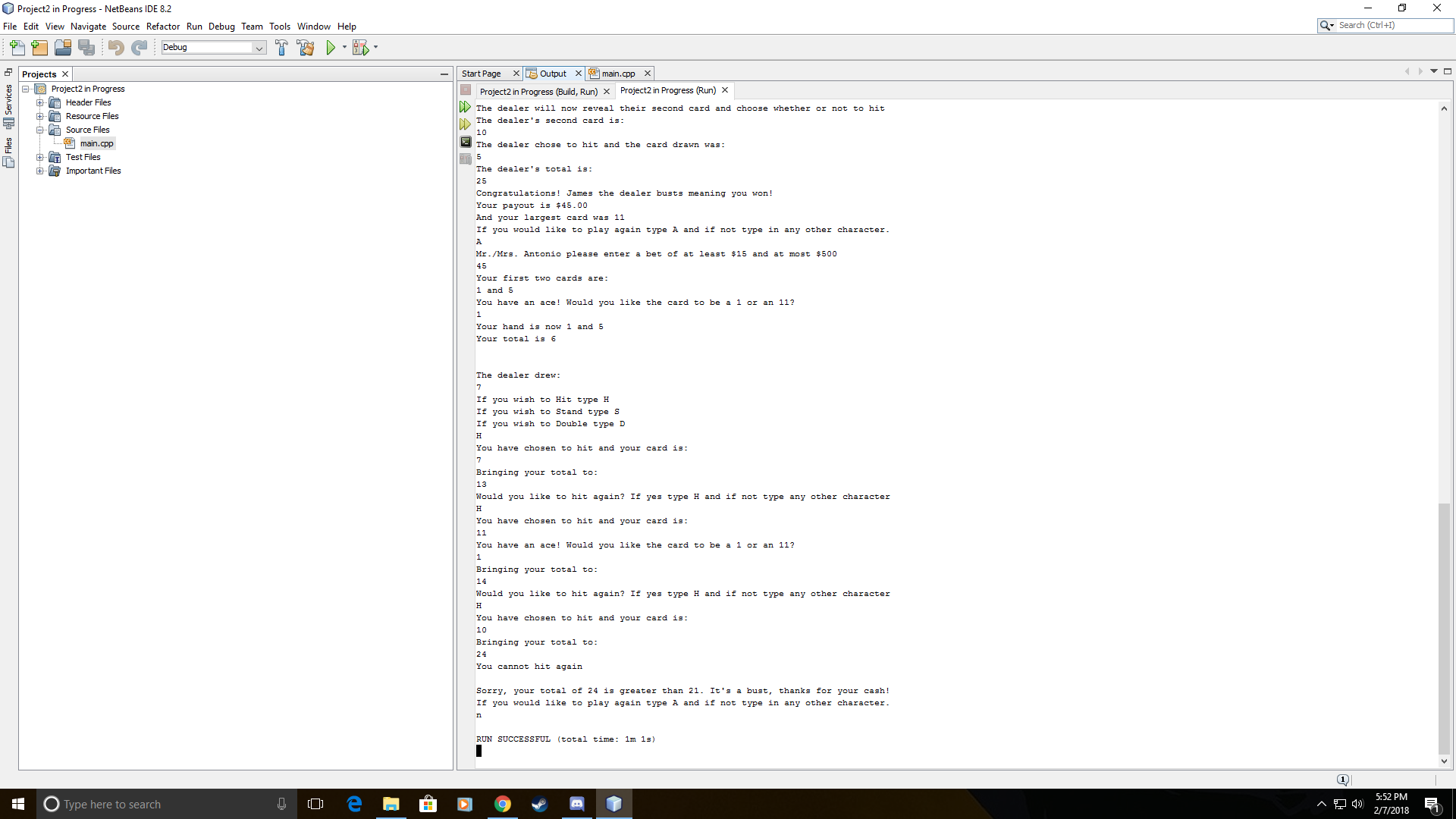
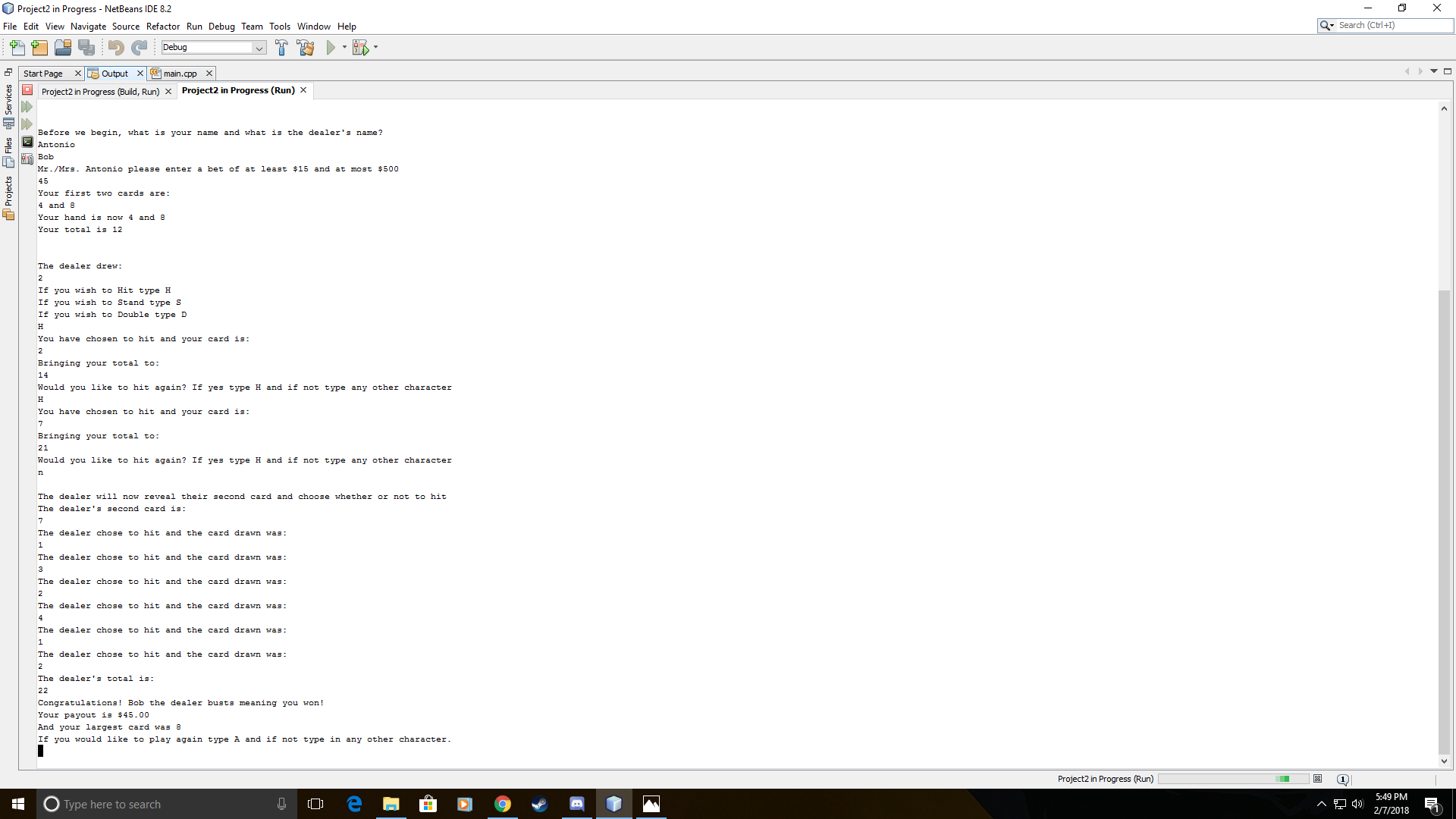
Size: 360 lines

It took me approximately 4 days to implement the new concepts (arrays, vectors, functions...etc.) and most of the difficulty was found in passing a 2-D array as a function. Once I learned how to pass a 2-D array into a function, I was able to finish the remainder of the project within the next day or so. Said array was used to hold all of the randomly generated card values. Beyond this there wasn’t much else in the way of difficulty for implementing the new concepts primarily because the foundations had been set in project 1.

**Proof of a Working Project**

In the case that the text is too small to see, all of these images will be included as JPEG’s in the project folder.





**Psuedo Code**

*Global Constants*

*Quantity of people = 2*

*Function Prototypes*

*Determine Ace*

*Player’s total cards Dealt*

*Array Cards Dealt to Player and Dealer*

*Record Player’s and Dealer’s Names*

*Sort the Dealer’s Cards with a Bubble Sort*

*Sort the Dealer’s Cards with a Select Sort*

*Largest Card in the Dealer’s Hand*

*Automatic Win if First hand is 21*

*Show Array after Both Sorts*

*Initalize*

*Explain the procedure of the game from file*

*Ask for name of player and dealer via Function*

*Do{*

*Call for Function to Fill Array Randomly*

*While both cards are the same and equal to eleven, re-randomize*

*Ask for player’s bet while it is out of bounds*

*Give player first two cards from array*

*If one or both cards is equal to 1 or 11 got to function to determine value of ace*

*Record the value of the greatest card for later*

*If the cards are equal to 21*

*The player automatically wins*

*Else Continue*

*Display one of dealer’s cards from array*

*Ask player’s what they want to do based on their hand*

*If they don’t have a split*

*Ask if they want to hit*

*Ask if they want to stay*

*Ask if they want to double*

*If they do have a split*

*Ask if they want to hit*

*Ask if they want to stay*

*Ask if they want to double*

*Ask if they want to split*

*Switch based on choice*

*Switch hit*

*Do*

*Pull card 3 from array based on value count*

*Increment count*

*If card 3 is equal to 1 or 11 call for Ace function*

*Compare value of card 3 to first 2 and store*

*Compare value of card 3 to value potential*

*Card 3 equal to potential*

*Add to the total value of the cards*

*Display card 3*

*Display total value of cards*

*If total is less than 21*

*Ask if they want to hit again*

*Input decision to hit*

*Else they cannot hit again*

*While the total is less than 21 and if they want to hit again*

*Break out of switch*

*Switch stay*

*Break out of switch*

*Switch double*

*Pull card 3 from array*

*Increment the bet and display the bet*

*If card 3 is equal to 1 or 11 call for Ace function*

*Display card 3*

*Display total value of cards*

*If card 3 is greater than card 2*

*Store value of card 3 for later*

*Else store the value of card 1*

*Break out of the switch*

*Switch split*

*Set card 1 equal to hand 1*

*Set card 2 equal to hand 2*

*Increment the bet*

*Pull card 3 from array*

*Pull card 4 from array*

*Deal the third card to the first hand*

*Display card 3 from Array*

*If card 3 is equal to 1 or 11 call ace function*

*If card 3 is greater than card 2*

*If card 3 is greater than card 4*

*Store the value of card 3 for later*

*Deal the fourth card card to the second hand*

*Display card 4 from array*

*If card 4 is equal to 1 or 11 call ace function*

*If card 4 is greater than card 2*

*If card 4 is greater than card 3*

*Store the value of card 4 for later\*

*Total of the first hand*

*Total of the second hand*

*If the first hand is greater than the second hand*

*The total that will be used is the first hand*

*Else the total that will be used is the second hand*

*Break out of switch*

*If total is greater than 21 it is a bust and you lost*

*Else*

*The dealer’s total is calculated*

*The dealer’s second card is pulled and revealed from an array*

*While the dealer’s total is less than the player’s*

*The dealer’s hit card is pulled from array and displayed*

*If the dealer’s hand is greater than 21*

*The dealer busts and you win*

*Display the payout*

*Display the highest card in your hand*

*Else If your hand is greater than the dealer’s and less than or equal to 21*

*You beat the dealer*

*Display payout*

*Display highest card in your hand*

*Else If the dealer’s hand is higher than the player’s*

*The dealer beat you*

*Display the dealer’s largest card from function*

*Sort array with bubble sort*

*Display Sorted Cards*

*Sort array with select sort*

*Display sorted cards*

*Display bet lost*

*Else If the total of both the player’s and the dealer’s hand is equal*

*You tied, nothing was lost*

*Would you like to play again?*

*Input decision*

*}while the decision is yes*

*Function that displays sorted arrays*

*From i equals 0 to i equals count +1*

*Disply sorted array*

*Function to determine Automatic Win based on First Hand*

*If number 1 + 2 = 21*

*val=true*

*Else value equal to false*

*Return Val*

*Function to determine dealer’s Largest Card*

*Large is equal to 0*

*For i equal to 0 to i =count +1*

*If the number in array is greater than previous greatest value it is equal to large*

*Return large*

*Function Selection Sort*

*Initialize Scan, MinIndx, MinVal*

*For scan equal to 0 scan less than num + 1*

*MinIndx=Scan*

*MinVal=Array at scan*

*For index=scan+1 index less than num +1*

*If array at index is less than MinVal*

*Min Val is equal to Array at index value*

*MinIndx is equal to index*

*Array at MinIndx is equal to Array at scan*

*Array at Scan is equal to MinVal*

*Function Bubble Sort*

*For i equal to 0 to i equal to num*

*For j=i + 1 to j equal to num + 1*

*If array at i is greater than array at j*

*Temp value equal to array at i*

*Array at i is equal to array at j*

*Array at j is equal to temp value*

*Function to Input Names*

*For i is equal to 0 to i is equal to PEOPLE*

*Input names*

*Function to Fill Array*

*For j is equal to 0 to j is equal to 1*

*For i is equal to 0 to i is equal to 10*

*Randomly generate values for array[i][j]*

*Function to Display Total*

*Total is equal to card 1 plus card 2*

*Return total*

*Function to Determine Value of an Ace*

*Input choice of either 1 or 11 for n*

*While n is not equal to 1 or 11*

*Re-enter value of n*

*Return n*

**Flowchart**

Due to its size it is in a separate file within the folder

**Code**

/\*

\* File: main.cpp

\* Author: Antonio Gines

\* Created on February,4 , 2018, 10:00 AM

\* Purpose: Blackjack 21

\*/

//System Libraries

#include <iostream>

#include <iomanip>

#include <cstdlib>

#include <ctime>

#include <cmath>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

//User Libraries

//Global Constants - Math/Physics Constants, Conversions,

// 2-D Array Dimensions

const int PEOPLE=2; //The amount of people in the match

//Function Prototypes

int ace(); //Choose Whether an Ace=1 or 11

int Dealt(int &,int &); //Player's Total Cards Dealt

void Dealt(int [][PEOPLE]); //Cards Dealt to the Dealer

void Names(vector<string> &,int); //Player's and Dealer's Name

void Dsort1(int [][PEOPLE], int); //Sort the Dealer's Cards

void Dsort2(int [][PEOPLE], int);

int Largest(int [][PEOPLE], int); //Dealer's Largest Card

bool AutoWin(int, int); //Determine if Automatic Win by 21

void ShowAry(int [][PEOPLE], int);

//Execution Begins Here

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

//Declare Variables

int SIZE=10;

int PPL=1;

int bet,newbet,total,rng1,rng2,card1,card2,card3,card4,payout,ptntial;

int total1,total2,split1,split2,dealer1,dealer2,dealer3,dealtot;

int dealer[10][2];

char choice, hitagn, playagn;

bool win;

vector<string> name(2); //Vector To be Used for Names

ifstream file;

//Seed Random Number Generator

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

//Initialize Variables

ptntial=0; //Will be used later on as temp

//Rules of the Game

file.open("Blackjack.txt"); //Open The file that gives instructions

string line;

while(file){

getline(file,line);

cout<<line<<endl;

}file.close(); //Close the file

//Begin Game

cout<<"Before we begin, what is your name and what is the "

<<"dealer's name?"<<endl;

Names( name, PPL); //Call FillAry

do{

int count1=0;

newbet=0;

bet=0;

Dealt(dealer); //Dealer's and Player's Randomized Cards

while(card1==11&&card1==card2){

card2=(rand()%11)+1;

}

do{ //Enter Player name

cout<<"Mr./Mrs. "<<name[0]<<" please enter a bet of at least $15 and at"

<<" most $500"<<endl;

cin>>bet;

}while(bet>500||bet<15);

card1=dealer[0][0];

card2=dealer[0][1];

cout<<"Your first two cards are:"<<endl;

cout<<card1<<" and "<<card2<<endl; //Deal The Cards

if(card1==1||card1==11){ //Choosing what happens if you get

card1=ace(); //1 or an 11

}if(card2==1||card2==11){

card2=ace();

}

cout<<"Your hand is now "<<card1<<" and "<<card2<<endl;

total=Dealt(card1, card2);

//Record Largest Card's Value

if(card1>card2){

rng1=card1;

}else rng1=card2;

//Determine Auto Win

win=AutoWin(card1, card2);

win?cout<<"Congratulations! It's a Blackjack,you win"<<endl:cout<<endl;

if(win){ //If it is 21

cout<<"Your payout is:"<<endl;

cout<<"$"<<fixed<<setprecision(2)<<showpoint<<static\_cast<float>(bet)

<<endl;

}else cout<<endl;

if(win){

exit(0);

}

cout<<"The dealer drew:"<<endl; //Dealer's Turn

cout<<dealer[1][0]<<endl;

if(card1==card2){ //Choices when you are able to split

cout<<"If you wish to Hit type H"<<endl;

cout<<"If you wish to Stand type S"<<endl;

cout<<"If you wish to Split type T"<<endl;

cout<<"If you wish to Double type D"<<endl;

}if(card1!=card2){ //Choices when you are unable to split

cout<<"If you wish to Hit type H"<<endl;

cout<<"If you wish to Stand type S"<<endl;

cout<<"If you wish to Double type D"<<endl;

}

cin>>choice;

while(choice!='H'&&choice!='S'&&choice!='T'&&choice!='D'){//Validating user input

cout<<"You have entered an invalid choice please try again"<<endl;

cin>>choice;

}if(choice=='T'&&card1!=card2){ //Inability to split when cards don't =

while(choice=='T')

cout<<"You cannot split because your cards are not the same. Please"

<<" enter a valid response"<<endl;

cin>>choice;

}

switch(choice){ //Begin switch based on input

case 'H':do{ //Choosing to hit

card3=dealer[0][count1+2];

count1++;

cout<<"You have chosen to hit and your card is:"<<endl;

cout<<card3<<endl;

if(card3==1||card3==11){ //Choosing what happens if you get

card3=ace(); //1 or an 11

}

total+=card3; //Incrementing total value of cards

if(card3>card2){

rng2=card3;

}else if(ptntial>card3&&ptntial>card2){

rng2=ptntial;

}else rng2=card1;

ptntial=card3;

cout<<"Bringing your total to:"<<endl;

cout<<total<<endl;

if(total<=21){ //Choosing to hit again

cout<<"Would you like to hit again? If yes type"

<<" H and if not type any other character"<<endl;

cin>>hitagn;

}else cout<<"You cannot hit again"<<endl;

}while(hitagn=='H'&&total<=21);

cout<<endl;break;

case 'S':cout<<endl;break; //Choosing to stay

case 'D':bet+=bet; //Choosing to double and incrementing bet

card3=dealer[0][2];

cout<<"You have doubled your initial bet to"<<endl;

cout<<fixed<<setprecision(2)<<showpoint

<<static\_cast<float>(bet)<<endl;

cout<<"You will be given another card and cannot hit again"

<<endl;

cout<<"Your card is"<<endl;

cout<<card3<<endl;

if(card3==1||card3==11){ //Choosing what happens if you get

card3=ace(); //1 or an 11

}

cout<<"And your total is"<<endl;

total+=card3; //Incrementing total value of cards

if(card3>card2){ //Saving input to determine max value

rng2=card3; //of single card in a hand

}else rng2=card1;

cout<<total<<endl;break;

case 'T':split1=card1; //Splitting the hand

split2=card2;

cout<<"You have doubled your initial bet and split it into "

<<"two hands."<<endl;

bet+=bet; //Incrementing the bet

card3=dealer[0][2];

card4=dealer[0][3];

cout<<"The dealer will now add a card to both hands."<<endl;

cout<<"To the first hand the dealer added"<<endl;

cout<<card3<<endl; //Card added to the first hand

if(card3==1||card3==11){ //Choosing what happens if you get

card3=ace(); //1 or an 11

}

if(card3>card2){ //Saving input

if(card3>card4){

rng2=card3;

}

}else rng2=card1;

cout<<"To the second hand the dealer added"<<endl;

cout<<card4<<endl; //card added to second hand

if(card4==1||card4==11){ //Choosing what happens if you get

card4=ace(); //1 or an 11

}

if(card4>card2){

if(card4>card3){ //Saving input

rng2=card4;

}

}else rng2=card1;

total1=card1+card3; //Total value of each hand

total2=card2+card4;

cout<<"The total of the first pile is"<<endl;

cout<<total1<<endl;

cout<<"The total of the second pile is"<<endl;

cout<<total2<<endl;

if(total1>total2){ //Determine which hand is

total=total1; //Favorable

}else total=total2;

cout<<endl;break;

}if(total>21){ //It's a bust if > 21

cout<<"Sorry, your total of "<<total<<" is greater than 21. It's a "

<<"bust, thanks for your cash!"<<endl;

}else{

dealtot=dealer[1][0]+dealer[1][1]; //If it's not an immediate bust

cout<<"The dealer will now reveal their second card and choose whether "

<<"or not to hit"<<endl;

cout<<"The dealer's second card is:"<<endl;

cout<<dealer[1][1]<<endl;

int count=0;

while(dealtot<=total){ //Determine whether dealer hits

dealtot+=dealer[1][count+2];

cout<<"The dealer chose to hit and the card drawn was:"<<endl;

cout<<dealer[1][count+2]<<endl;

count++;

}if (count<=1){

count=1;

}

cout<<"The dealer's total is:"<<endl;

cout<<dealtot<<endl;

if(dealtot>21){ // If Dealer Busts

cout<<"Congratulations! "<<name[1]

<<" the dealer busts meaning you won!"<<endl;

cout<<"Your payout is $"<<fixed<<setprecision(2)<<showpoint

<<static\_cast<float>(bet)<<endl;

cout<<"And your largest card was "

<<static\_cast<int>(fmax(rng1,rng2))<<endl;

}

else if(total<=21&&total>dealtot){ //If you win

cout<<"Congratulations! You beat "<<name[1]<<" the dealer! "<<endl;

cout<<"Your payout is $"

<<fixed<<setprecision(2)<<showpoint

<<static\_cast<float>(bet)<<endl;

cout<<"And your largest card was "

<<static\_cast<int>(fmax(rng1,rng2))<<endl;

}else if(dealtot>total){ //If the dealer busts

int large=Largest(dealer, count);

Dsort1(dealer, count); //Bubble Sorts

cout<<name[1]<<"'s cards organized using bubble sort were:"<<endl;

ShowAry(dealer, count);

Dsort2(dealer, count); //Selection Sorts

cout<<name[1]<<"'s cards organized using selection sort were:"

<<endl;

ShowAry(dealer, count);

cout<<endl;

cout<<"Sorry, but "<<name[1]

<<" the dealer beat you and you lost $"<<fixed

<<setprecision(2)<<showpoint<<static\_cast<float>(bet)<<endl;

cout<<"The dealer's largest card was "<<large<<endl;

}else if(dealtot==total){ //If you tie

cout<<"You and the dealer tied so you get to keep your $"<<fixed

<<setprecision(2)<<showpoint<<static\_cast<float>(bet)<<endl;

}

}

cout<<"If you would like to play again type A and if not type in"

<<" any other character."<<endl;

cin>>playagn; //Would you like to replay?

}while(playagn=='A'||playagn=='a'); //Choose to replay

//Exit stage right!

return 0;

}

void ShowAry(int Array[][2], int number){ //Display Both Sorted Arrays

for(int i=0;i<=number+1;i++){

cout<<Array[1][i]<<" ";

}cout<<endl;

}

bool AutoWin(int num1, int num2){ //If first hand is 21

bool Val;

if(num1+num2==21){

Val=true;

}else Val=false;

return Val;

}

int Largest(int Array[][2], int nums){ //Largest Card in hand

int large=0;

for(int i=0;i<=nums+1;i++){

if(Array[1][i]>large){

large=Array[1][i];

}

}return large;

}

void Dsort2(int Array[][2], int nums){ //Selection Sort

int Scan, MinIndx, MinVal;

for(Scan=0;Scan<=nums-1;Scan++){

MinIndx=Scan;

MinVal=Array[1][Scan];

for(int index=Scan+1;index<=nums;index++){

if (Array[1][index]<MinVal){

MinVal=Array[1][index];

MinIndx=index;

}

}

Array[1][MinIndx]=Array[1][Scan];

Array[1][Scan]=MinVal;

}

}

void Dsort1(int SortAry[][2], int num){ //Bubble Sort

for(int i=0;i<=num;i++){

for(int j=i+1;j<=num+1;j++){

if(SortAry[1][i]>SortAry[1][j]){

int temp=SortAry[1][i]; //Pass By Reference

SortAry[1][i]=SortAry[1][j];

SortAry[1][j]=temp;

}

}

}

}

void Names(vector<string> &name, int people){ //Input Names

for(int i=0;i<=people;i++){

cin>>name[i];

}

}

void Dealt(int array[][2]){ //Fill An Array

int deal;

for(int j=0;j<=1;j++){

for(int i=0;i<=10;i++){

array[i][j]=(rand()%11)+1;

}

}

}

int Dealt(int &card1,int &card2){ //Display Total

int T;

T=card1+card2;

cout<<"Your total is "<<T<<endl;

return T;

}

int ace(){ //Determine if it is an Ace

int n;

cout<<"You have an ace! Would you like the card to be a 1 or an 11?"<<endl;

cin>>n;

while(n!=1&&n!=11){

cout<<"You have not entered a valid number, please try again"<<endl;

cin>>n;

}

return n;

}

**Check-off Sheet**

The check-off sheet is in a separate file within the folder. NOTE: It may be distorted due to the transition from ods to word dox and I did not intend for this distortion to happen