

PROJECT 1

<MONKEY GAME>

CSC17A - 42636
Cay, Shienne Patricia
Date: 04/17/2017

Introduction

Title: Monkey Game

Monkey Game is played using a deck of cards. It can have players up to 4 people but in this case, only 2 players are accommodated. The one with the highest point after the end of the game wins, and the loser is called the Monkey of the game.

The game goes when the machine divides the shuffled cards to two players so each player holds 26 cards. The player will then discard all the 'pairs' in their hand to gather points. Pairs are the same face value of card disregarding its suit. So, if player one has a '4S' and a '4C', it is a pair and must be discarded. The player gets two points each time a pair is discarded. They get a point deduction if they input two cards that are not a pair. After the end of this round, if no one got deduction, players must have even points. If a player has not seen a pair in hand and proceeded to end his/her turn, he/she will have the option to discard later, although it will only be one point.

After all elimination, players will face each other's cards and try to guess which is a pair of their card in the opponent's hand. The player will choose a card in hand, and an opponent card. If it is a pair, 3 points is added to the player, and he/she gets another turn. If it's not a pair, no point deduction happens. It will simply be the next player's turn. If cards have not been eliminated after five turns of each players, game ends and machine will tally the score. Announcement of winner will be made and recorded to the "Records" text file.

How It Works

Once user hit play, the program will explain the mechanics of the game. The user will be prompted to play the game or not.

Should the user choose not to play, game ends and exits. If the user decides to play, player will be asked if he/she choose to be Player 1 or Player 2 of the game. Whoever the user chooses to be, he/she will have the first turn.

```
Welcome to the Monkey Game!

Mechanics: There are two players in the game and the cards are
evenly divided between the two. If a player has the same value
in one hand, the two card goes into the bin. The goal is to be
the first to lose all the cards. When there are no more of the
the pair, each player gets to pick one card from the opponent.
Do this until one player wins. Loser is called the MONKEY!

Would you like to play the game?
    Y - Yes | N - No

CHOICE: █
```

```
PLAY GAME AS:

    1 - Player 1 | 2 - Player 2
      Q - QUIT

CHOICE: █
```

There is still option to quit game in choosing players should the player wants to end game. Once user selects a player, the machine will prompt user for the names of

two players for recording purposes. The names are not used until the end of the game. After inputting the names for both players, machine will notify that cards have been dispersed and depending on the user's player preference, the user's card will be shown while the opponent's cards are hidden.

```
CHOICE: 1

Player 1 (Name): Shienne
Player 2 (Name): Patricia

Cards have been dispersed!

PLAYER 1: QC 7D 4H KC 5H 5D 5C 7S TS 4S 9C AC 6C JD 2D JC QD TC AD 2S JH TD KD QS 3D 6S
          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

PLAYER 2: ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** 
          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Press ENTER to start game!
█
```

The game will immediately start once user presses 'Enter' key.

```
PLAYER 1's Turn

PLAYER 1: QC 7D 4H KC 5H 5D 5C 7S TS 4S 9C AC 6C JD 2D JC QD TC AD 2S JH TD KD QS 3D 6S
          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 1
2nd Card: 24

You got it!
```

This loop will continue until player 1 sees no more pair and inputs 0 in both cards. And there it becomes player 2's turn.

```
Player 2's Turn

PLAYER 2: KH JS 8D AS 4C 2H 9D 7C 3C QH TH 9H 9S 2C 5S 8C 6D 8S 6H AH KS 8H 3S 3H 7H 4D
          1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

Note: To quit entire game, input the same number for 1st card & 2nd card.
If there's no more pair available, input '0' in both cards. To eliminate
pairs in your hand, put in the corresponding number of your card. If the
input cards are not a pair, you get a point deduction. Otherwise, if the
cards are a pair, you get 2 points!

Put in the corresponding number of your cards!

1st Card: 5
2nd Card: 26

You got it!
```

When players are done in eliminating pairs in their hands, score board will be shown and game will ask if player wants to discard a pair if ever they missed a pair in first elimination.

```

=====
                        SCORE BOARD
=====
PLAYER 1 ---- 20 pts.
PLAYER 2 ---- 18 pts.
=====

Player 1's Turn

PLAYER 1: AC JD 5D 6D 3C KH
          10 14 20 22 23 26

PLAYER 2: ** ** ** ** ** ** ** ** 
          5  6  7 12 14 18 25 26

Would you like to discard a pair in your hand?
          Y - Yes | N - No
CHOICE: n

```

Eliminated pair after first elimination only gives the player a point instead of two. After this, player will guess a possible pair of his/her own hand to the opponent's cards. Game will determine if it's a pair. If it is, player continues to play. If not, it becomes the turn of the next player.

The game also displays the turns left for each player in guessing round. Each player starts with overall 5 turns. Once it is all used up, the game ends and the program will tally the score. Then it will announce the winner.

```

PLAYER 1: TD 4C JS 3S 7S 8D
          8 13 14 16 22 23

PLAYER 2: ** ** ** 
          7 12 15 20 23 24

Note: Try to eliminate card in your hands as you pick one card
      from your opponent. If it's a pair, you get 3 points, and you
      still get another turn. No deductions if your wrong. If cards
      are not discarded in 5 turns, game ends and score tallies!

Reminder: Input '0' in both cards to quit!

Turns Left: 5

Your Card: 8
Opponent's Card: 20

It's not a pair!

```

```

PLAYER 1: 7S 2C
PLAYER 2: 7C 2S

=====
                        SCORE BOARD
=====
PLAYER 1 ---- 36 pts.
PLAYER 2 ---- 18 pts.
=====

Shienne WINS!

Data recorded to a file.

```

If game ended after 5 turns are used up, the game will display the cards left and announce winner. If the game ended because a player got it all right, game skips card display.

Additional Photos/Screenshots

Player 1's Turn

PLAYER 1: JH AH 8S 7H 3H 8C 2H KD
6 14 15 18 19 20 22 25

PLAYER 2: ** ** * * * *
3 6 11 15 20 22

Would you like to discard a pair in your hand?
Y - Yes | N - No

CHOICE: y

Input corresponding number.

1st Card: 15

2nd Card: 20

You got it!

Would you like to play the game?

Y - Yes | N - No

CHOICE: 4

You can only choose Y or N!

PLAY GAME AS:

1 - Player 1 | 2 - Player 2
Q - QUIT

CHOICE: 5

You can only choose 1, 2 or Q!

PLAYER 1: ** ** * * * *
8 13 16 22 23

PLAYER 2: TC 8C 3C 4S 7H
7 12 15 20 23

Note: Try to eliminate card in your hands as you pick one card from your opponent. If it's a pair, you get 3 points, and you still get another turn. No deductions if your wrong. If cards are not discarded in 5 turns, game ends and score tallies!

Reminder: Input '0' in both cards to quit!

Turns Left: 4

Your Card: 23

Opponent's Card: 22

You got it!

Summary

Project Size: 820 lines

The Number of Variables: about 30

I have applied memory allocation when I allocated data from file for each card, functions with structures used as input and output, and pointers with array of structures. I also used a bit of character array for player names. I read from a file and wrote to it but I have not used binary files as it has no need for this game. Also, I was not able to validate if user puts in a character in a supposedly number input. Other than that, the menus have validation.

This took me a week to make as it is full of do-while loops – nested and stand-alone – for the game and most of the functions are pass by reference. I applied what I learned in previous class, CSC5, such as Boolean operators and Boolean functions which turned out to be so useful in this project.

I had a hard time blocking out cards that are already chosen since if I sort the array, it is already finalized and no way to add and delete. To make it work, every time I pass the array that contains all eliminated cards, I made a copy array available only in the function and I sort that out instead of the original array so I could still add eliminated cards as the game progresses. Every time I call the function, it makes a copy so it's like a new array sorted for display.

Description

The main point I programmed in this project is formatting cards in each players' hand. I had to find out how to make discarding cards work without messing up the whole program. I also had to find a way to have every value in array start at 1 instead of 0 to make discarding possible.

Cross Reference for Project 1

Chapter	Section	Topic	Where in code – Line Number
9	2	Pointer Variables	77
	5	Initializing Pointers	77
	7	Pointers as function argument	36, 38
	8	Dynamic Memory Allocation	426
	9	Returning Pointers from Functions	430
10	3	C-strings stores as array	65, 124, 126
11	2	Combining data into structures	Array.h
	3	Accessing structure members	409, 419, 415 ...
	5	Arrays of Structures	407-419
	7	Structures as function argument	39-46, 49-52, 56
	8	Returning structure from function	36, 402, 422
	9	Pointers to structures	409, 415, 419...
12	1	File Operations	128, 366
	3	Passing File Stream objects to functions	36, 407

Pseudo-code

```
/*  
 * File:  main.cpp  
 * Author: Shienne Cay  
 * Created on April 17, 2017, 9:20 PM  
 * Purpose: Monkey Game  
 */  
  
//System Libraries  
  
//User Libraries  
  
//Global Constants  
//Such as PI, Vc, -> Math/Science values  
//as well as conversions from one system of measurements  
//to another  
  
//Function Prototypes  
//Fill structure array  
//Fill index pointer  
//Shuffle index for cards  
//Delete allocated memory  
//Show Card of each with other player hidden  
//Show Card of each with other player hidden  
//Show cards of both players  
//Validate user input if it is a pair  
//Validate user input if it is a pair part 2  
//Sort temporary array  
//Points so far  
//Both cards left after removing pairs  
//Player 1 Cards left  
//Player 2 cards left  
//Both Players cards left  
//Check for validation  
//Check for validation  
//Check if all cards are discarded  
//Show cards for both players without hidden  
  
//Executable code begins here! Always begins in Main  
  
//Set the random number seed
```

```

//Declare Variables

//Explain rules

//Prompt user for game play
    //If they chose to play, get to next loop
    //If 'n', delete index pointer, end game
    //Invalidate if input isn't 'Y' or 'N'

//Prompt user to choose player
    //If they chose 1, become player 1
    //Exit loop
    //If they chose 2, become player 2
    //Exit loop
    //If they chose 'Q', delete index. end game
    //Invalidate if input '1', '2', or 'Q'

//Prompt user for name (player 1 & 2)
//Filestream in
    //Fill deck from cards in file

//Show player his/her card

//Start first part of the game - loop hand card elimination
    //If each player had their turn, exit elimination loop
        //If player 1, show player 1 deck
        //If Player 2, show player 2 deck

//Start elimination loop
    //Prompt user for first card
    //Prompt user for second card
        //If any input is less than 0 or greater than 26, invalidate!
        //If user put '0' in both card, exit elimination loop
    //If player 1, change to player 2
        //Add game play counter
    //If player 2, change to player 1
        //Add game play counter
    //Set a to false
    //If input is the same except 0, quit game play
        //Delete deck and index
    //If input is none other than above, check player
        //If player 1
            //Check if input has already been used

```

```

        //If not, validate if cards are a pair
        //If cards are a pair, add to trash array
        //Add two points for player
        //Show Player Cards
        //If cards are not a pair
        //Deduct 1 point
    //If input has already been used
        //Invalidate and show player cards
    //Do the same if player is player 2
//Continue loop if a remains true
//Continue loop if change remains true

//Show Score board

//If no turns left, end game loop
    //If player is player 1
    //Show cards deck
    //If player is player 2
    //Show cards deck

//Ask if user wants to discard a pair in hand
    //If yes, prompt entering the two cards
        //If player 1, check if input number is within hand
            //If it is, validate if both cards are a pair
            //If yes, add cards to trash array
            //Add only 1 point for pairing
            //Show cards left
        //If invalid, deduct 1 point to score
        //If input number is not within hand,
            //invalidate. show cards left
    //If no, exit discarding loop
//If no more cards left, exit all loop
//If cards still exist, show cards

//Explain rules for 2nd part
//Show turns left based on player
//Prompt user to input card number from hand
//Prompt user to input number from opponent's hand
    //If player chose to quit, end game straight to tally
    //If player 1, check if input is in both hands
        //If yes, validate if it is a pair
        //If it is, add both to trash bin
        //Add 3 points to player

```

```
//Continue game
//If not a pair, change player
//Deduct a turn from player 1
//Exit inner loop
//If input is unidentified, notify user
//If player 2, do the same with player 1

//Continue loop if c remains true
//Continue loop if end remains true

//Show score board after game
//Write results to a file

//If player 1 has highest point, declare player 1 winner with name embedded
//If player 2 has highest point, declare player 2 winner with name embedded
//Announce if it's a tie
//Write every result to a file
//Close file

//Delete structure array
//Delete index array

//End game
```

Program

```
/*
 * File:  main.cpp
 * Author: Shienne Cay
 * Created on April 11, 2017, 9:20 PM
 * Purpose: Monkey Game
 */

//System Libraries
#include <iostream>
#include <fstream>
#include <ctime>
#include <cstdlib>
#include <iomanip>
using namespace std;

//User Libraries
#include "Array.h"

//Global Constants
//Such as PI, Vc, -> Math/Science values
//as well as conversions from one system of measurements
//to another

//Function Prototypes
Card *filDeck (int, int *, fstream &, string);
int *index (int);
void shuffle (int *, int);
void destroy(Card *);
void shwCrd1 (Card *, char);
void shwCrd2 (Card *, char);
void shwBoth (Card *, char);
void shwPly1 (Card *, int [], int);
void shwPly2 (Card *, int [], int);
bool valid8 (Card *, short, short, bool);
bool valid82 (Card *, short, short, bool);
void srtNum (int [], int);
void pntSfar (int, int);
void bothLft (Card *, int [], int [], bool, int);
void ply1Lft (Card *, bool, int [], int);
void ply2Lft (Card *, bool, int [], int);
void shwPlay (Card *, bool, int [], int [], int);
bool check1 (int [], short, short, int);
bool check2 (int [], int [], bool, short, short, int);
bool finish (int [], int[], int);
void shwAll (Card *, int [], int [], int);

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

```

srand(static_cast<unsigned int>(time(0)));

const int NSIZE=80;
char player1[NSIZE], player2[NSIZE],
    start, player, discard;
int size=52, p1pnt=0, p2pnt=0,
    gmPly1=0, gmPly2=0, cntr1=1,
    cntr2=1;
int trsh1[27]={ }, trsh2[27]={ };
bool x=true, y=true,
    a=true, b=true, c=true,
    change=true, play=true,
    end=true;
short frst, scnd;
short trn1=5, trn2=5;
int *stand=index(size);
shuffle(stand, size);

cout<<"\t\tWelcome to the Monkey Game!"<<endl<<endl;
cout<<"Mechanics: There are two players in the game and the cards are\n"
    <<"evenly divided between the two. If a player has the same value\n"
    <<"in one hand, the two card goes into the bin. The goal is to be\n"
    <<"the first to lose all the cards. When there are no more of the\n"
    <<"the pair, each player gets to pick one card from the opponent.\n"
    <<"Do this until one player wins. Loser is called the MONKEY!"<<endl<<endl;

do {
    cout<<"Would you like to play the game?"<<endl;
    cout<<"    Y - Yes | N - No"<<endl;
    cout<<"\nCHOICE: ";
    cin>>start;
    if (start=='y' || start=='Y') x=false;
    else if (start=='n' || start=='N') {
        delete []stand;
        return 0;
    }
    else cout<<"\nYou can only choose Y or N!"<<endl<<endl;
} while(x);

do {
    cout<<"\n          PLAY GAME AS:"<<endl<<endl;
    cout<<"    1 - Player 1 | 2 - Player 2"<<endl;
    cout<<"          Q - QUIT"<<endl;
    cout<<"\nCHOICE: ";
    cin>>player;
    if (player=='1') {
        play=true;
        y=false;
    }
    else if (player=='2') {

```

```

        play=false;
        y=false;
    }
    else if (player=='Q' || player=='q') {
        delete []stand;
        return 0;
    }
    else cout<<"\nYou can only choose 1, 2 or Q!"<<endl<<endl;
} while(y);

cin.ignore(256, '\n');
cout<<"\nPlayer 1 (Name): ";
cin.getline(player1, NSIZE);
cout<<"Player 2 (Name): ";
cin.getline(player2, NSIZE);

fstream in;
Card *deck=filDeck(size, stand, in, "deck.txt");
cout<<"\nCards have been dispersed!"<<endl;

shwBoth(deck, player);
cout<<"Press ENTER to start game!"<<endl;
cin.get();

do {
    if (gmPly1==1 && gmPly2==1) change=false;
    else {
        if (play) {
            cout<<"\nPLAYER 1's Turn"<<endl<<endl;
            shwPly1(deck, trsh1, 27);
        }
        else {
            cout<<"\nPlayer 2's Turn"<<endl<<endl;
            shwPly2(deck, trsh2, 27);
        }
        do {
            cout<<"Note: To quit entire game, input the same number for 1st card & 2nd card."<<endl;
            cout<<"If there's no more pair available, input '0' in both cards. To eliminate"<<endl;
            cout<<"pairs in your hand, put in the corresponding number of your card. If the"<<endl;
            cout<<"input cards are not a pair, you get a point deduction. Otherwise, if the"<<endl;
            cout<<"cards are a pair, you get 2 points!"<<endl<<endl;
            cout<<"Put in the corresponding number of your cards!"<<endl<<endl;
            cout<<"1st Card: ";
            cin>>frst;
            cout<<"2nd Card: ";
            cin>>scnd;
            if (frst<0 || frst>26 || scnd<0 || scnd>26) {
                cout<<"\nNumber unidentified!"<<endl<<endl;
                if (play) shwPly1(deck, trsh1, 27);
                else shwPly2(deck, trsh2, 27);
            }
        }
    }
} while (change);

```

```

}
else {
    if (frst==0&&scnd==0) {
        if (play) {
            gmPly1++;
            play=false;
        }
        else {
            gmPly2++;
            play=true;
        }
        a=false;
    }
    else if (frst==scnd) {
        destroy(deck);
        delete []stand;
        return 0;
    }
    else {
        if (play) {
            if (check1(trsh1, frst, scnd, 27)) {
                if (valid8(deck, frst, scnd, play)) {
                    trsh1[ctr1]=frst; ctr1++;
                    trsh1[ctr1]=scnd; ctr1++;
                    cout<<"\nYou got it!"<<endl;
                    p1pnt+=2;
                    shwPly1(deck, trsh1, 27);
                }
                else {
                    cout<<"\nIt's not a pair!"<<endl;
                    cout<<"1 point is deducted to your score!"<<endl<<endl;
                    p1pnt--;
                }
            }
            else {
                cout<<"\nNumber unidentified!"<<endl<<endl;
                shwPly1(deck, trsh1, 27);
            }
        }
        else {
            if (check1(trsh2, frst, scnd, 27)) {
                if (valid8(deck, frst, scnd, play)) {
                    trsh2[ctr2]=frst; ctr2++;
                    trsh2[ctr2]=scnd; ctr2++;
                    cout<<"\nYou got it!"<<endl;
                    p2pnt+=2;
                    shwPly2(deck, trsh2, 27);
                }
                else {
                    cout<<"\nIt's not a pair!"<<endl;
                    cout<<"1 point is deducted to your score!"<<endl<<endl;

```



```

        p2pnt--;
    }
}
else {
    cout<<"\nNumber unidentified!"<<endl<<endl;
    shwPly2(deck, trsh2, 27);
}
}
}
} while(a);
} while(change);

pntSfar(p1pnt, p2pnt);
do {
    if (trn1==0&&trn2==0) end=false;
    else {
        if (play) {
            cout<<"Player 1's Turn"<<endl<<endl;
            shwPlay(deck, play, trsh1, trsh2, 27);
        }
        else {
            cout<<"Player 2's Turn"<<endl<<endl;
            shwPlay(deck, play, trsh1, trsh2, 27);
        }
        do {
            cout<<"\nWould you like to discard a pair in your hand?"<<endl;
            cout<<"          Y - Yes | N - No          "<<endl;
            cout<<"CHOICE: ";
            cin>>discard;
            if (discard=='Y' || discard=='y') {
                cout<<"\nInput corresponding number."<<endl<<endl;
                cout<<"1st Card: ";
                cin>>frst;
                cout<<"2nd Card: ";
                cin>>scnd;
                if (play) {
                    if (check1(trsh1, frst, scnd, 27)) {
                        if (valid8(deck, frst, scnd, play)) {
                            trsh1[cntr1]=frst; cntr1++;
                            trsh1[cntr1]=scnd; cntr1++;
                            cout<<"\nYou got it!"<<endl;
                            p1pnt++;
                            ply1Lft(deck, play, trsh1, 27);
                        }
                    }
                    else {
                        cout<<"\nIt's not a pair!"<<endl;
                        cout<<"1 point is deducted to your score!"<<endl<<endl;
                        p1pnt--;
                    }
                }
            }
        }
    }
}

```

```

    }
    else {
        cout<<"\nNumber unidentified!"<<endl<<endl;
        ply1Lft(deck, play, trsh1, 27);
    }
}
else {
    if (check1(trsh2, frst, scnd, 27)) {
        if (valid8(deck, frst, scnd, play)) {
            trsh2[ctr2]=frst; ctr2++;
            trsh2[ctr2]=scnd; ctr2++;
            cout<<"\nYou got it!"<<endl;
            p2pnt++;
            ply2Lft(deck, play, trsh2, 27);
        }
        else {
            cout<<"\nIt's not a pair!"<<endl;
            cout<<"1 point is deducted to your score!"<<endl<<endl;
            p2pnt--;
        }
    }
    else {
        cout<<"\nNumber unidentified!"<<endl<<endl;
        ply2Lft(deck, play, trsh2, 27);
    }
}
}
else if (discard=='N' || discard=='n') {                b=false;
}
else cout<<"\nYou can only choose Y or N!"<<endl<<endl;
} while (b);
c=true;
do {
    if (finish(trsh1, trsh2, 27)) {
        end=false;
        c=false;
    }
    else {
        shwPlay(deck, play, trsh1, trsh2, 27);
        cout<<"\nNote: Try to eliminate card in your hands as you pick one card"<<endl;
        cout<<" from your opponent. If it's a pair, you get 3 points, and you"<<endl;
        cout<<" still get another turn. No deductions if your wrong. If cards"<<endl;
        cout<<" are not discarded in 5 turns, game ends and score tallies!"<<endl<<endl;
        cout<<"Reminder: Input '0' in both cards to quit!"<<endl<<endl;
        if (play) cout<<"Turns Left: "<<trn1;
        else cout<<"Turns Left: "<<trn2;
        cout<<"\n\nYour Card: ";
        cin>>frst;
        cout<<"Opponent's Card: ";
        cin>>scnd;
        if (frst==0 && scnd==0) {

```

```

        c=false;
        end=false;
    }
    else {
        if (play) {
            if (check2(trsh1, trsh2, play, frst, scnd, 27)) {
                if (valid82(deck, frst, scnd, play)) {
                    trsh1[ctr1]=frst; ctr1++;
                    trsh2[ctr2]=scnd; ctr2++;
                    cout<<"\nYou got it!"<<endl;
                    p1pnt+=3;
                }
                else {
                    cout<<"\nIt's not a pair!"<<endl<<endl;
                    play=false;
                    trn1--;
                    c=false;
                }
            }
            else cout<<"\nNumber unidentified!"<<endl<<endl;
        }
        else { //If player 2, do the same with player 1
            if (check2(trsh1, trsh2, play, frst, scnd, 27)) {
                if (valid82(deck, frst, scnd, play)) {
                    trsh2[ctr2]=frst; ctr2++;
                    trsh1[ctr1]=scnd; ctr1++;
                    cout<<"\nYou got it!"<<endl;
                    p1pnt+=3;
                }
                else {
                    cout<<"\nIt's not a pair!"<<endl<<endl;
                    trn2--;
                    play=true;
                    c=false;
                }
            }
            else cout<<"\nNumber unidentified!"<<endl<<endl;
        }
    }
} while(c);
} while (end);

if (trn1==0&&trn2==0) shwAll(deck, trsh1, trsh2, 27);
pntSfar(p1pnt, p2pnt);

fstream out;
out.open("Records.txt", ios::out | ios::app);

```

```

//Output values
if (p1pnt>p2pnt) {
    cout<<"\t\t"<<player1<<" WINS!"<<endl<<endl;
    cout<<"Data recorded to a file."<<endl;
    out<<player1<<" WINS!"<<endl<<endl;
    out<<player1<<" - "<<p1pnt<<" points"<<endl;
    out<<player2<<" - "<<p2pnt<<" points"<<endl<<endl;
}
else if (p1pnt<p2pnt) {
    cout<<"\t\t"<<player2<<" WINS!"<<endl<<endl;
    cout<<"Data recorded to a file."<<endl;
    out<<player2<<" WINS!"<<endl<<endl;
    out<<player2<<" - "<<p2pnt<<" points"<<endl;
    out<<player1<<" - "<<p1pnt<<" points"<<endl<<endl;
}
else {
    cout<<"\t\t"<<"IT'S A TIE!"<<endl<<endl;
    cout<<"Data recorded to a file."<<endl;
    out<<"IT'S A TIE!"<<endl<<endl;
    out<<player1<<" - "<<p1pnt<<" points"<<endl;
    out<<player2<<" - "<<p2pnt<<" points"<<endl<<endl;
}
out.close();

destroy(deck);
delete []stand;
return 0;
}

```