

Project 1

<Concentration Memory Game>

CIS5/CSC5 40404

Name: Wuysang, Emma

Date: 1/30/2024

Introduction

Title: Concentration Card Game

This is a memory-based/guessing game.

The goal is to find as many pairs as possible, with the least moves possible.

With this project implementation, it will be catered to finding one of the pairs, out of all twenty pairs. A half a stack of cards, only the suit of Hearts and Spades will be shuffled (excluding Joker and Royals). 20 Cards (Ace-9) will be laid out individually.

A matching pair is defined as two cards with the same rank. (two Aces, two fives, etc.) Since there isn't a visual stack of cards, the user must type a placement (Exa. a b, c f), after choosing, the cards will be revealed.

Summary

Project size: about 480 lines

The number of variables: about 77

Includes the 7 constructs learned.

The project took way more lines than it should have, and since we had a limited number of constructs(etc.) that we learned, I tried to utilize what I knew. I have always been enamored by memory-based games, and I remember growing up, I would play this game with a group of friends.

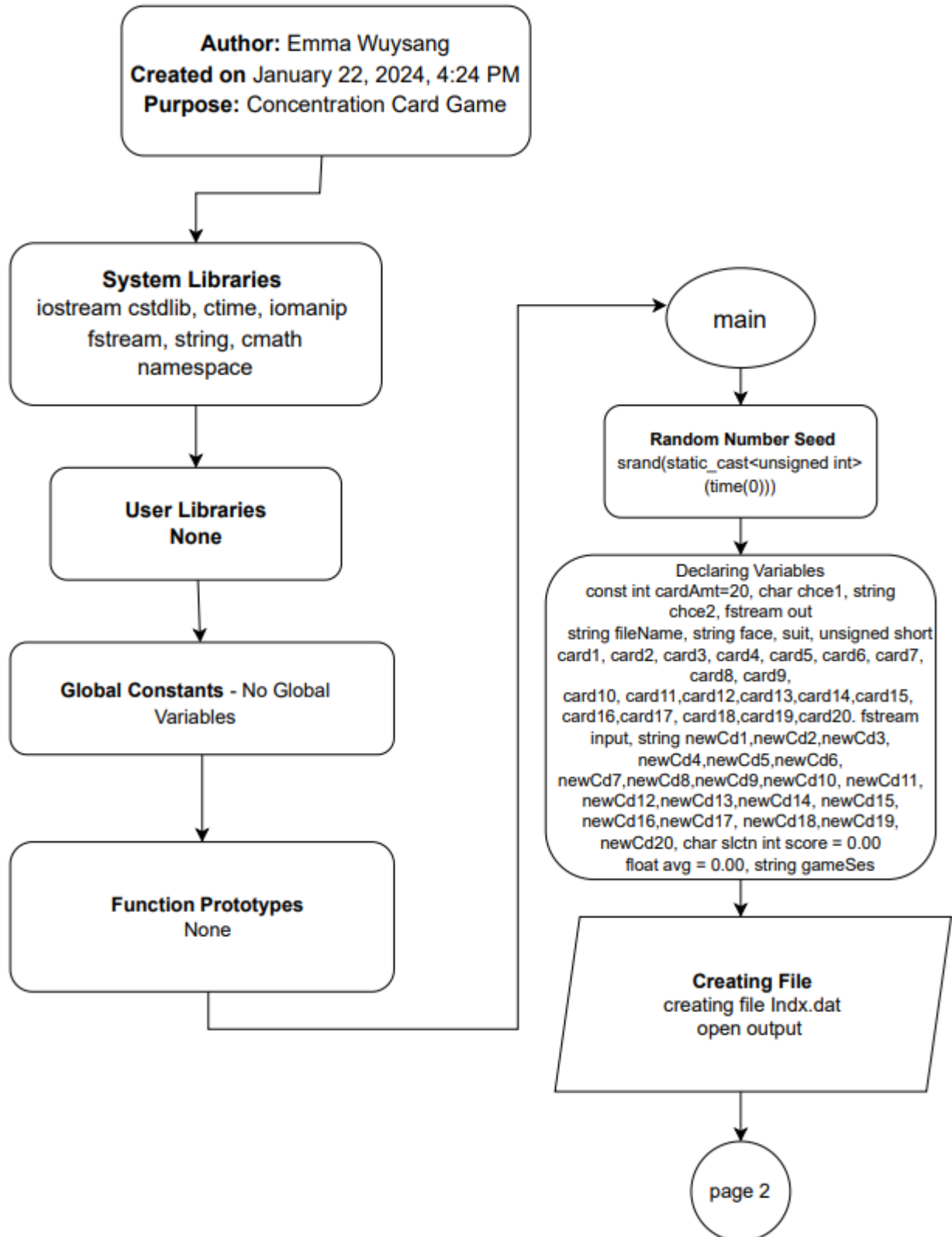
It took around a few days, and mainly hours within those days. I would have had more versions if I did not spend long periods of time working on the project. Continuously around 3-9 hours a day for around four days, I spent reviewing old material and trying to implement what I knew. I had no previous C++ experience, which made it a bit difficult to take in as much information.

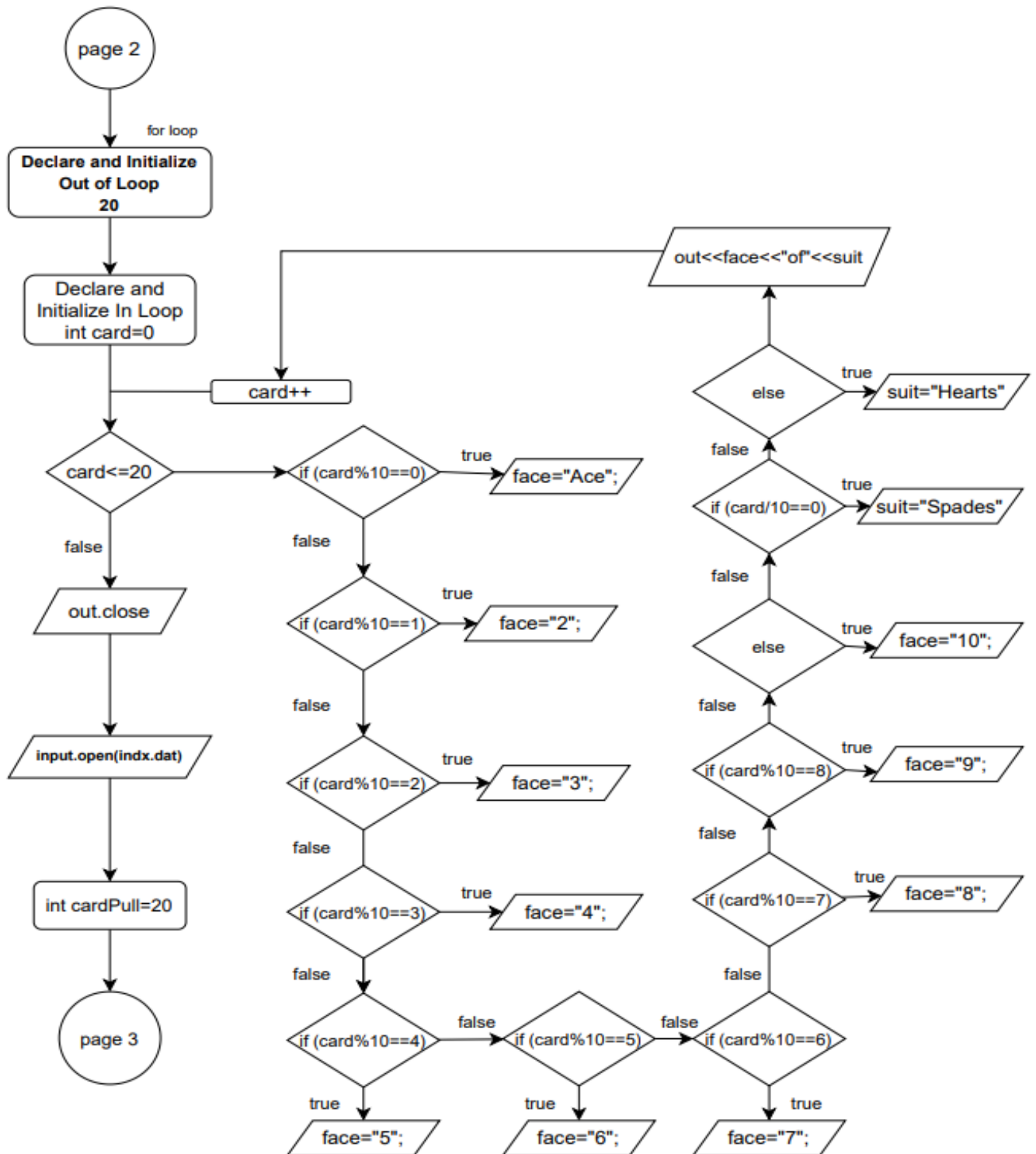
It was fun; if there were more time as a normal semester session, this project would have felt less constructed, but I did as much as I could within the week.

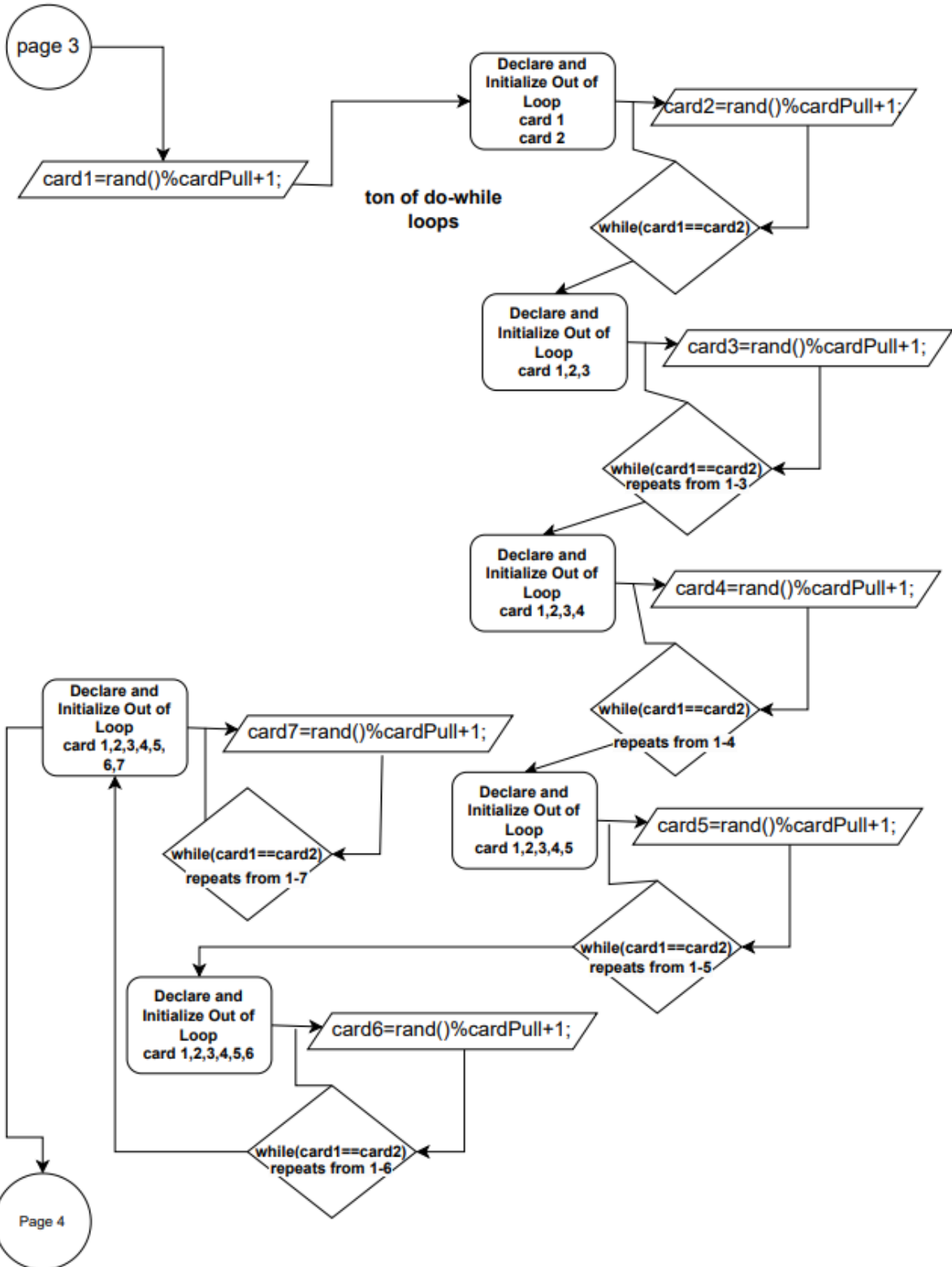
Description

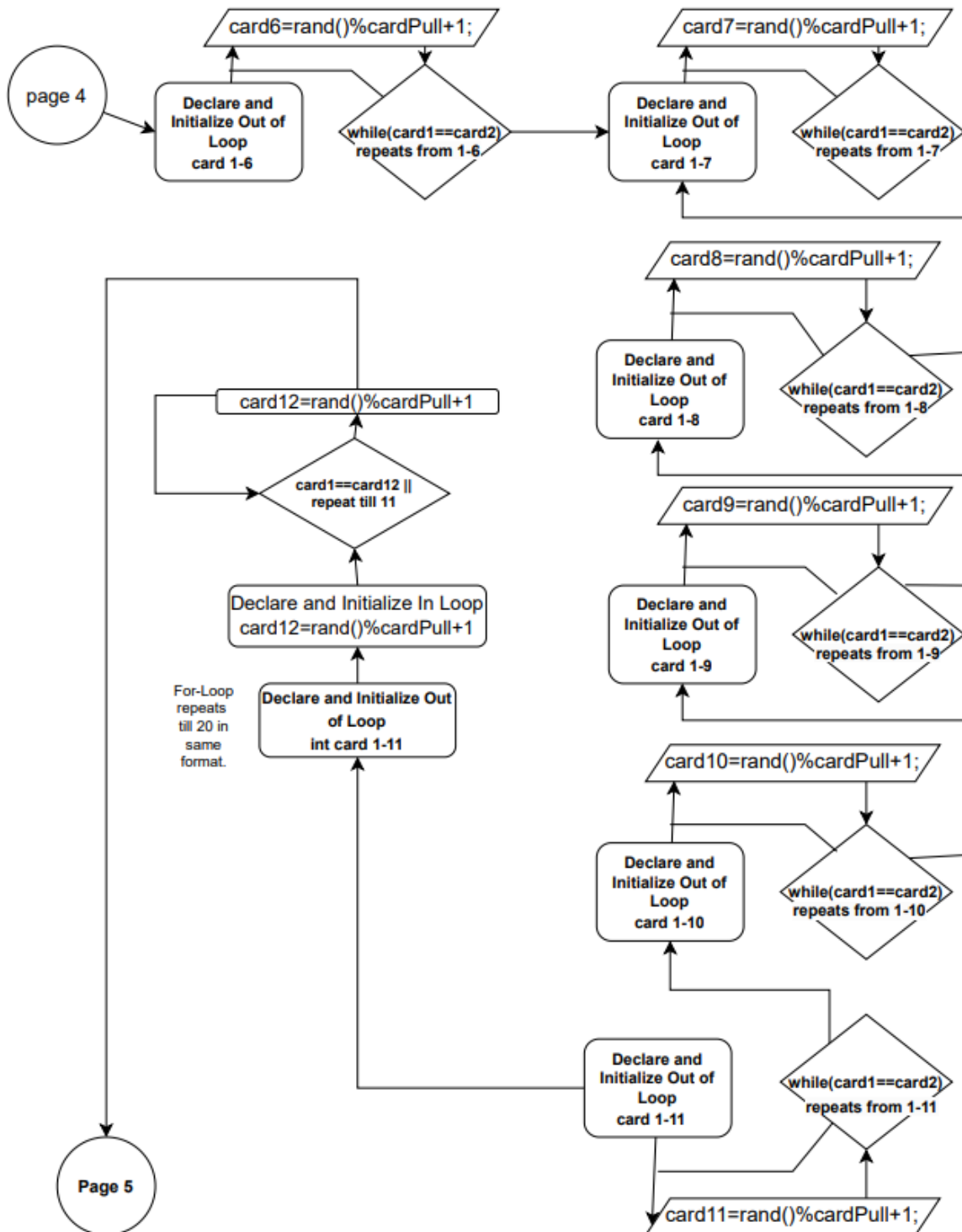
The main point is to match letter a to one of the 19 choices in the least amount of moves possible; despite it not being fully concentration, I wanted to implement the concept.

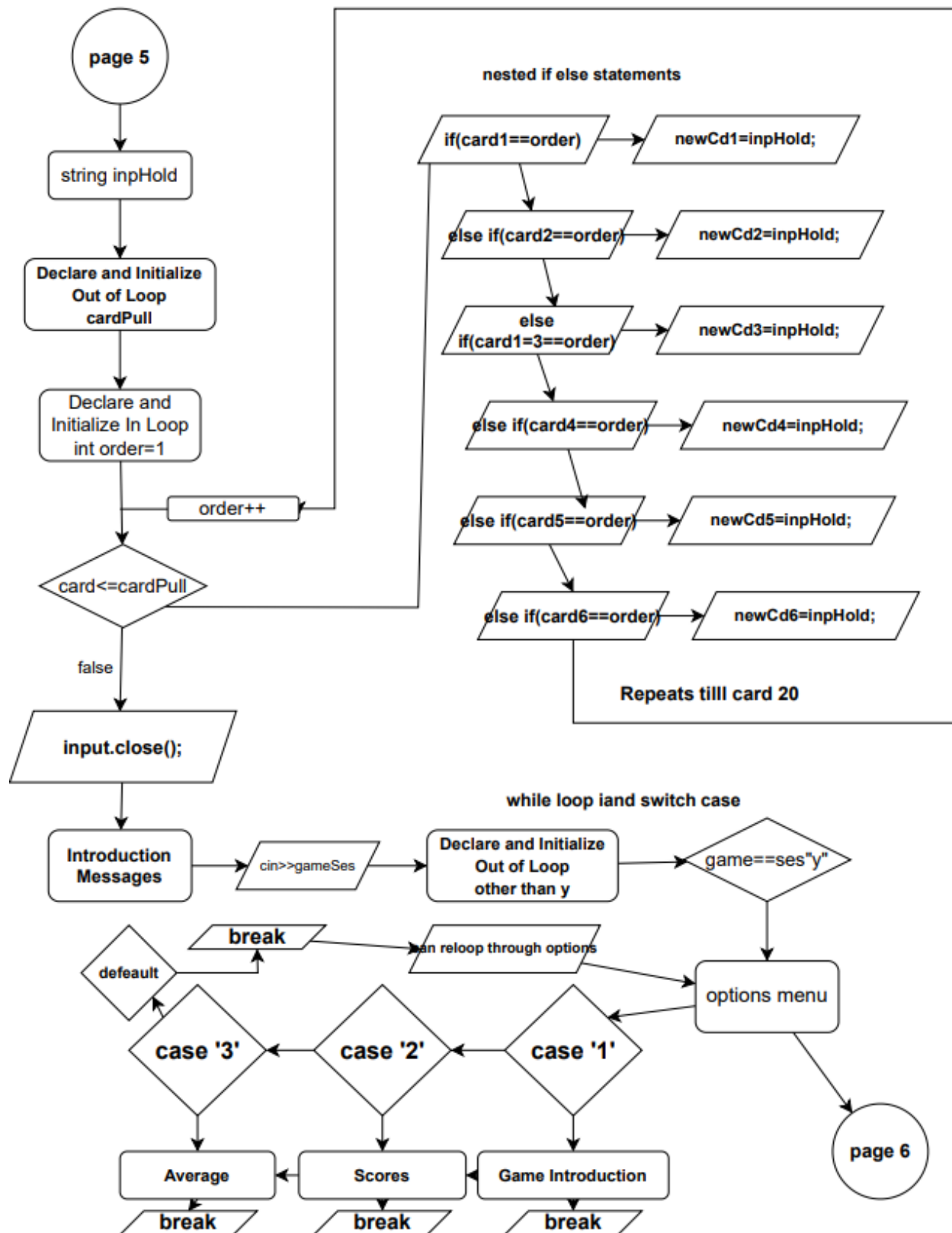
Flowchart

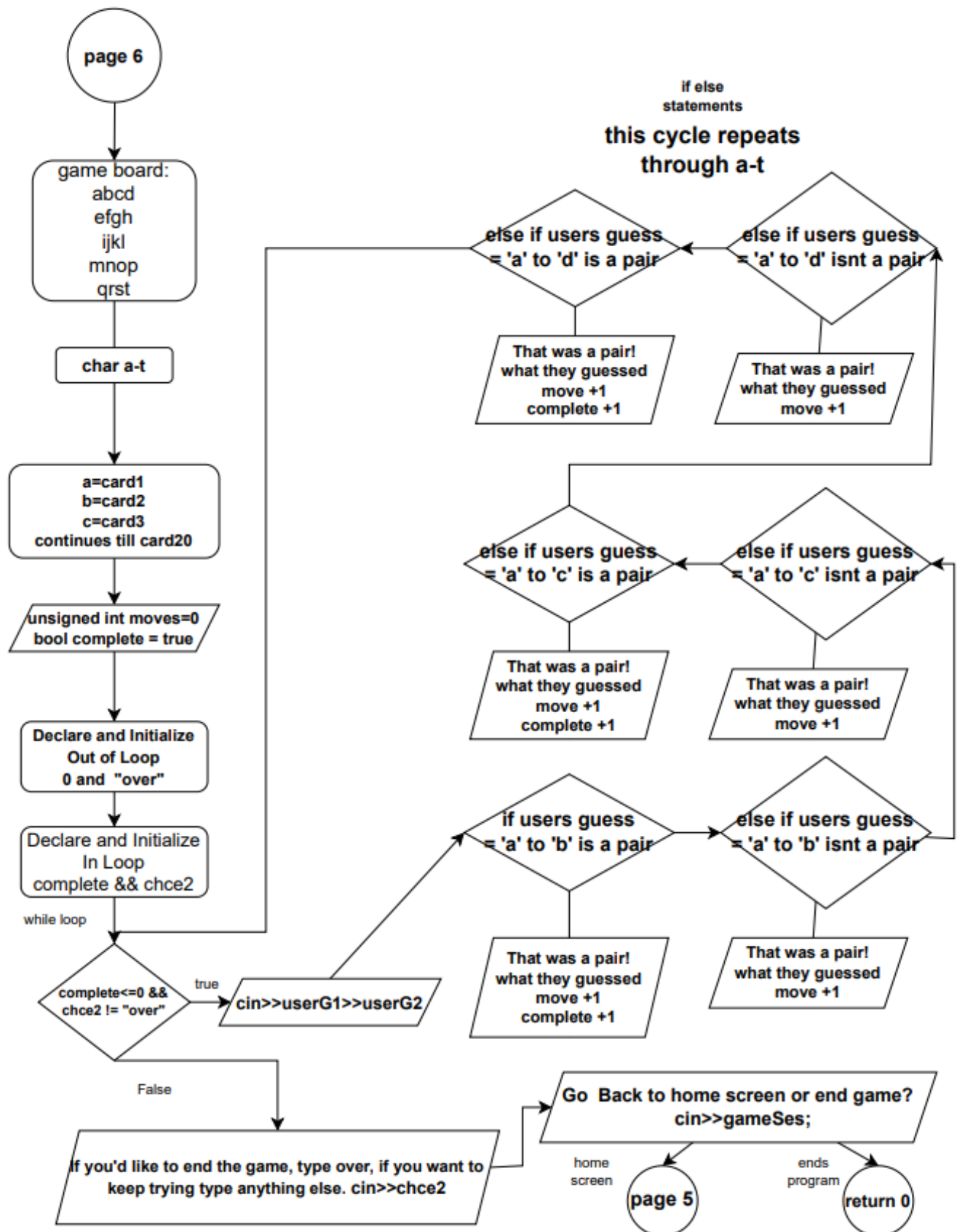












Pseudo Code

```
// Pseudocode for Concentration Card Game
```

```
// Import necessary libraries
```

```
// Declare global constants and variables
```

```
// Initialize random seed  
Utilize Srand from Ctime
```

```
// Open and write card index to file  
Open file 'Indx.dat' for writing  
for each card from 1 to 20.
```

```
Determine face and suit of the card  
Write face + "of" + suit to the file  
Close the file
```

```
// Open the card index file for reading  
Open file 'Indx.dat' for reading
```

```
// Initialize card variables  
card1, card2, ..., card20  
For 20 cards, each time, go through file, and assign the random placement to a string.
```

```
Randomly select unique cards from the index file, signify a shuffle  
// ... (repeat for cards 2 to 20)  
Close the file
```

```
// Display introduction and options  
Display "Welcome to the Concentration Card Game!"  
Prompt user to enter the loading screen (y/n)
```

```
while user wants to continue since game session is yes  
    Display options menu  
    Get user's choice (slctn)
```

```
switch slctn
```

```

    case '1': // Introduction
        Display game introduction
    case '2': // Scores
        Display current score, this is the amount of moves
    case '3': // Average
        Calculate and display average guessing rate
    default:
        // Do nothing or exit loop
Keep switch working till user types something else
end switch

```

end while

```

// Display game board and prompt to start playing
Prompt user to start the game (chcel)

```

While the user wants to play the user said they want to play and game isn't decided to be over, keep going.

```

    Display game board
Match letter a to the correct pair!
    Get user's two letter guesses (userG1, userG2)
    Check each letter to the letter A/a

    // Check if the guesses form a pair and update the game state
    if (userG1, userG2) forms a pair then
        Display "That was a pair!"
Set complete to 1 if guessed right
    Display cards that form the pair
    Display number of moves and update the score
Tell user if the input isn't a pair, along with the cards
end if the user says its over

```

```

    Prompt user to continue playing or exit the game
    Get user's choice (chcel)
Either go back to home screen or end program->
end while

```

Cross Reference for Project 1

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #'s	Pts	Notes
2	2	cout	220,238-244,275		
	3	libraries	all 8 included 9-15 275,227,266,113,98,184,27	8	iostream, iomanip, cmath, cstdlib, fstream, string, ctime
	4	variables/literals	42,43, 314, 101		No variables in global area, failed project!
	5	Identifiers	30-44		
	6	Integers	314,42,30	3	
	7	Characters	290,41,31	3	
	8	Strings	39,34,35,44,185	3	
	9	Floats No Doubles	43	3	Using doubles will fail the project, floats OK!
	10	Bools	315,45	4	
	11	Sizeof *****			
	12	Variables 7 characters or less	30-44		All variables <= 7 characters
	13	Scope ***** No Global Variables			
	14	Arithmetic operators	102,113,264		
	15	Comments 20%+	all throughout code	5	Model as pseudo code
	16	Named Constants	30		All Local, only Conversions/Physics/Math in Global area
	17	Programming Style ***** Emulate			Emulate style in book/in class repository
3	1	cin	222, 233, 276, 317		
	2	Math Expression	264		
	3	Mixing data types ****			
	4	Overflow/Underflow ****			
	5	Type Casting	27, 264	4	
	6	Multiple assignment *****			
	7	Formatting output	284,285,286,287,288	4	
	8	Strings	321,329,465,469	3	
	9	Math Library	266	4	All libraries included have to be used
	10	Hand tracing *****			
4	1	Relational Operators	320,226,273		
	2	if	248, 83	4	Independent if
	4	If-else	253,258	4	
	5	Nesting	277-478. 252-256	4	
	6	If-else-if	324,328,332	4	
	7	Flags *****			
	8	Logical operators	320,324,277,324	4	
	11	Validating user input	285, 477, 481,277	4	
	13	Conditional Operator	324,328,332,283	4	
	14	Switch	240	4	
5	1	Increment/Decrement	318-474	4	
	2	While	225,277,285,320	4	
	5	Do-while	115,120,125,130	4	
	6	For loop	50,158,161,170	4	
	11	Files input/output both	49, 99,217,95	8	
	12	No breaks in loops *****	none (only in switch)		Failed Project if included
*****	Not required to show		Total	100	

Reference

1. Textbook
2. Lehr's GitHub Repository
3. [Concentration Game Introduction/How to Play](#)

Versions

Version 1: The bare bones and beginning of my project for the memory game. The issue with the way I was starting this program was first, being a bit confused about the game. I knew there was a memory game, but I could not pinpoint until a few days later, that it was also a card game. So, within version one, there are a lot of output statements, not much programming.

Version 2: With programming for version 2, I set out to create a shuffle to the card. I knew I had to figure out how to get shuffled set of numbers to grab out of the file, and that was my main priority. Version 2 was successful in doing so; however, the matching was not actually made yet. Added tad bits of game like implementations (options, home screen, many choices, etc.)

Version 3: This was probably the longest I had spent on this project. Version 3 was close to the final draft of what I wanted to upload. The issue was that midway through coding letters to matching each other 19 letters, it would be tedious when including all comparisons. I decided to switch the concept of the game to find the pair for a.

Version 4: The last day I spent programming this project, and mostly my work was done doing comments and fixing the loops. There were a few issues within version 3 that started to pop up the more I did the runs. This was more focused on refining and trying to fix what I could. Even though my code was unnecessarily long, I fixed what I could within the time frame. All my work was spaced out throughout the week until version 3-4. There are all the comment fixes and a few additionally added game options.

Input and Output Testcases

Run 1:

```
>> This the Concentration Card Game!
>> Would you like to enter to the loading screen?(y/n)
y
Welcome to the Options Menu!
Option One (type 1) : Introduction.
Option Two (type 2) : Scores.
Option Three (type 3) : Average.

Enter a Character to Skip Options Screen.
y

Want To Play?(y/any other character)

y

>> The Concentration Card Board!
| a b c d |
| e f g h |
| i j k l |
| m n o p |
| q r s t |

This is your board, try to find the matching pair to a!
Type two letters to see if you found a match! (exa. a g)
ah
Not a pair.
You guessed the 5ofSpades and the 6ofSpades.
sa
Not a pair.
You guessed the 7ofHearts and the 5ofSpades.
ai
That was a pair!
You guessed the 5ofSpades and 5ofHearts.
It took you 3 move(s) to find the pair!

If you'd like to end the game, type over, if you want to keep trying type anything else.
Reminder, most, if not each time you play the game, it'll be different! (Just re-run the game!)
over
Are you completely done? If you'd like to go back to the home screen type y, if not, any other character will do. y

Welcome to the Options Menu!
Option One (type 1) : Introduction.
Option Two (type 2) : Scores.
Option Three (type 3) : Average.

Enter a Character to Skip Options Screen.
You've played the game already!
2

Your score is currently 3.
Proj 1, V.1 WIP (more added in project 2)

Would you like to select another option? (type 1,2,3/ if not any type any other character.)
3

Your average guessing rate is currently 0.15.
Rounded, it is 0.00!
Proj 1, V.1 WIP (more added in project 2)

Would you like to select another option? (type 1,2,3/ if not any type any other character.)
no
Want To Play?(y/any other character)
```

Are you completely done? If you'd like to go back to the home screen type y, if not, any other character will do. yes

RUN SUCCESSFUL (total time: 2m 57s)

Run 2:

The Concentration Card Board!

	a	b	c	d	
	e	f	g	h	
	i	j	k	l	
	m	n	o	p	
	q	r	s	t	

This is your board, try to find the matching pair to a!

Type two letters to see if you found a match! (exa. a g)

ah

Not a pair.

You guessed the 10ofSpades and the 6ofSpades.

ha

Not a pair.

You guessed the 6ofSpades and the 10ofSpades.

ia

Not a pair.

You guessed the AceofSpades and the 10ofSpades.

al

Not a pair.

You guessed the 10ofSpades and the 4ofHearts.

an

That was a pair!

You guessed the 10ofSpades and 10ofHearts.

It took you 5 move(s) to find the pair!

If you'd like to end the game, type over, if you want to keep trying type anything else.

Reminder, most, if not each time you play the game, it'll be different! (Just re-run the game!)

over

Are you completely done? If you'd like to go back to the home screen type y, if not, any other character will do. done

Run 3:

The Concentration Card Board!

	a	b	c	d	
	e	f	g	h	
	i	j	k	l	
	m	n	o	p	
	q	r	s	t	

This is your board, try to find the matching pair to a!
Type two letters to see if you found a match! (exa. a g)
ad
Not a pair.
You guessed the 10ofSpades and the 6ofHearts.
da
Not a pair.
You guessed the 6ofHearts and the 10ofSpades.
ag
Not a pair.
You guessed the 10ofSpades and the 3ofSpades.
am
Not a pair.
You guessed the 10ofSpades and the AceofSpades.
an
Not a pair.
You guessed the 10ofSpades and the 5ofSpades.
ao
Not a pair.
You guessed the 10ofSpades and the 4ofSpades.
ap
That was a pair!
You guessed the 10ofSpades and 10ofHearts.
It took you 7 move(s) to find the pair!

Program

```
/*  
 * File:  main.cpp  
 * Author: Emma Wuysang  
 * Created on January 28, 2024, 9:13 PM  
 * Purpose: Concentration Card Game V.4  
 */  
  
//System Libraries  
#include <iostream> //I/O Library  
#include <cstdlib> //Random Function Library  
#include <ctime> //Time Library  
#include <iomanip> //Formatting Library  
#include <fstream> //File Library  
#include <string> //String Objects  
#include <cmath> //Math Library  
using namespace std; //Library Name-space  
  
//User Libraries  
  
//Global Constants - Math,Physics,Chemistry,Conversions  
  
//Function Prototypes  
  
//Program Execution Begins Here  
int main(int argc, char** argv) {  
    //Set a random seed  
    srand(static_cast<unsigned int>(time(0))); // seeds w/ current time.  
  
    //Declare all variables
```



```

const int cardAmt=20; // Amount of Cards
char chce1; // choice 1
string chce2; // choice 2
fstream out; // output file
string fileName; // opens fstream
string face, suit; // Hearts and Spades + Ace-10
unsigned short card1, card2, card3, card4, card5, card6, card7, card8, card9, card10,
               card11,card12,card13,card14,card15,card16,card17,card18,card19,card20; // group of cards
fstream input; // input file
string newCd1,newCd2,newCd3,newCd4,newCd5,newCd6,newCd7,newCd8,newCd9,newCd10,
       newCd11,newCd12,newCd13,newCd14,newCd15,newCd16,newCd17,newCd18,newCd19,newCd20; // takes
the file, sets random sections to string.
char slctn; // selection
int score = 0.00; // general score
float avg = 0.00; // guessing rate
string gameSes; // go to loading screen/continue game
bool ran = false; // initializes

//Process or Map solutions
fileName="Indx.dat"; // card Index
out.open(fileName,ios::out);
for (int card=0;card<=20;card++){
    if (card%10==0){ // Ace
        face = "Ace";
    }
    else if (card%10==1){ // Two
        face = "2";
    }
    else if (card%10==2){ // Three
        face = "3";
    }
    else if (card%10==3){ // Four

```

```

        face = "4";
    }
    else if (card%10==4){ // Five
        face = "5";
    }
    else if (card%10==5){ // Six
        face = "6";
    }
    else if (card%10==6){ // Seven
        face = "7";
    }
    else if (card%10==7){ // Eight
        face = "8";
    }
    else if (card%10==8){ // Nine
        face = "9";
    }
    else{ // Ten
        face = "10";
    }

    // Spades Appending Suit
    if (card/10==0){
        suit="Spades";
    }
    // Hearts Appending Suit
    else{
        suit="Hearts";
    }

    out<<face<<" of "<<suit<<endl;
}

```

```
// close the file
out.close();

// file name
fileName="Indx.dat";
input.open(fileName.c_str(),ios::in);

//Initialize Variables
int cardPull=20;

//Unique Value for card 1.
card1=rand()%cardPull+1;
do{
    card2=rand()%cardPull+1;

//Unique Value for card 1,2.
}while(card1==card2);

//Unique Value for card 1,2,3.
do{
    card3=rand()%cardPull+1;
}while(card1==card3 || card2==card3);

//Unique Value for card 1,2,3,4.
do{
    card4=rand()%cardPull+1;
}while(card1==card4 || card2==card4 || card3==card4);

//Unique Value for card 1,2,3,4,5.
do{
    card5=rand()%cardPull+1;
```

```

}while(card1==card5 || card2==card5 || card3==card5 || card4==card5);

//Unique Value for card 1,2,3,4,5,6.
do{
    card6=rand()%cardPull+1;
}while(card1==card6 || card2==card6 || card3==card6 || card4==card6 || card5==card6);

//Unique Value for card 1,2,3,4,5,6,7.
do{
    card7=rand()%cardPull+1;
}while(card1==card7 || card2==card7 || card3==card7 || card4==card7 || card5==card7 || card6==card7);

//Unique Value for card 1,2,3,4,5,6,7,8.
do{
    card8=rand()%cardPull+1;
}while(card1==card8 || card2==card8 || card3==card8 || card4==card8 || card5==card8 || card6==card8 ||
card7==card8);

//Unique Value for card 1,2,3,4,5,6,7,8,9.
do{
    card9=rand()%cardPull+1;
}while(card1==card9 || card2==card9 || card3==card9 || card4==card9 || card5==card9 || card6==card9 ||
card7==card9 || card8==card9);

//Unique Value for card 1,2,3,4,5,6,7,8,9,10.
do{
    card10=rand()%cardPull+1;
}while(card1==card10 || card2==card10 || card3==card10 || card4==card10 || card5==card10 ||
card6==card10 || card7==card10 || card8==card10 || card9==card10);

//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11.
do{
    card11=rand()%cardPull+1;

```

```
}while(card1==card11 || card2==card11 || card3==card11 || card4==card11 || card5==card11 ||  
card6==card11 || card7==card11 || card8==card11 || card9==card11 || card10==card11);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12 using for loops.
```

```
for(card12=rand()%cardPull+1;card1==card12 || card2==card12 || card3==card12 || card4==card12 ||  
card5==card12 || card6==card12 || card7==card12 || card8==card12 || card9==card12 || card10==card12 ||  
card11==card12;card12=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13 using for loops.
```

```
for(card13=rand()%cardPull+1;card1==card13 || card2==card13 || card3==card13 || card4==card13 ||  
card5==card13 || card6==card13 || card7==card13 || card8==card13 || card9==card13 || card10==card13 ||  
card11==card13 || card12==card13;card13=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14 using for loops.
```

```
for(card14=rand()%cardPull+1;card1==card14 || card2==card14 || card3==card14 || card4==card14 ||  
card5==card14 || card6==card14 || card7==card14 || card8==card14 || card9==card14 || card10==card14 ||  
card11==card14 || card12==card14 || card13==card14;card14=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 using for loops.
```

```
for(card15=rand()%cardPull+1;card1==card15 || card2==card15 || card3==card15 || card4==card15 ||  
card5==card15 || card6==card15 || card7==card15 || card8==card15 || card9==card15 || card10==card15 ||  
card11==card15 || card12==card15 || card13==card15 || card14==card15 ;card15=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16 using for loops.
```

```
for(card16=rand()%cardPull+1;card1==card16 || card2==card16 || card3==card16 || card4==card16 ||  
card5==card16 || card6==card16 || card7==card16 || card8==card16 || card9==card16 || card10==card16 ||  
card11==card16 || card12==card16 || card13==card16 || card14==card16 ||  
card15==card16;card16=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17 using for loops.
```

```
for(card17=rand()%cardPull+1;card1==card17 || card2==card17 || card3==card17 || card4==card17 ||  
card5==card17 || card6==card17 || card7==card17 || card8==card17 || card9==card17 || card10==card17 ||  
card11==card17 || card12==card17 || card13==card17 || card14==card17 || card15==card17 ||  
card16==card17;card17=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18 using for loops.
```

```
for(card18=rand()%cardPull+1;card1==card18 || card2==card18 || card3==card18 || card4==card18 ||  
card5==card18 || card6==card18 || card7==card18 || card8==card18 || card9==card18 || card10==card18 ||
```

```
card11==card18 || card12==card18 || card13==card18 || card14==card18 || card15==card18 || card16==card18  
|| card17==card18;card18=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19 using for loops.
```

```
for(card19=rand()%cardPull+1;card1==card19 || card2==card19 || card3==card19 || card4==card19 ||  
card5==card19 || card6==card19 || card7==card19 || card8==card19 || card9==card19 || card10==card19 ||  
card11==card19 || card12==card19 || card13==card19 || card14==card19 || card15==card19 || card16==card19  
|| card17==card19 || card18==card19;card19=rand()%cardPull+1);
```

```
//Unique Value for card 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20 using for loops.
```

```
for(card20=rand()%cardPull+1;card1==card20 || card2==card20 || card3==card20 || card4==card20 ||  
card5==card20 || card6==card20 || card7==card20 || card8==card20 || card9==card20 || card10==card20 ||  
card11==card20 || card12==card20 || card13==card20 || card14==card20 || card15==card20 || card16==card20  
|| card17==card20 || card18==card20 || card19==card20 ;card20=rand()%cardPull+1);
```

```
//Pulls cards from the file
```

```
string inpHold; // input holder
```

```
/*loops from 1-20, taking the random num and finding
```

```
* the position of the string, finally,
```

```
* setting into a variable.
```

```
*/
```

```
for(int order=1;order<=cardPull;order++){
```

```
    input>>inpHold;
```

```
    if(card1==order)newCd1=inpHold;
```

```
    else if(card2==order)newCd2=inpHold;
```

```
    else if(card3==order)newCd3=inpHold;
```

```
    else if(card4==order)newCd4=inpHold;
```

```
    else if(card5==order)newCd5=inpHold;
```

```
    else if(card6==order)newCd6=inpHold;
```

```
    else if(card7==order)newCd7=inpHold;
```

```
    else if(card8==order)newCd8=inpHold;
```

```
    else if(card9==order)newCd9=inpHold;
```

```

else if(card10==order)newCd10=inpHold;
else if(card11==order)newCd11=inpHold;
else if(card12==order)newCd12=inpHold;
else if(card13==order)newCd13=inpHold;
else if(card14==order)newCd14=inpHold;
else if(card15==order)newCd15=inpHold;
else if(card16==order)newCd16=inpHold;
else if(card17==order)newCd17=inpHold;
else if(card18==order)newCd18=inpHold;
else if(card19==order)newCd19=inpHold;
else if(card20==order)newCd20=inpHold;
}

```

```

input.close(); // closes the file after taking in the variables

```

```

//Display the output

```

```

cout<<"This the Concentration Card Game!\n";
cout<<"Would you like to enter to the loading screen?(y/n) \n";
cin>>gameSes;

```

```

while(gameSes=="y"){ // leads to loading screen

```

```

    cout<<"_____ \n";
    cout<<setw(30)<<"Welcome to the Options Menu!\n"
    <<setw(31)<<"Option One (type 1) : Introduction. \n"
    <<setw(30)<<"Option Two (type 2) : Scores.\n"
    <<setw(30)<<"Option Three (type 3) : Average.\n"
    <<"_____ \n"
    <<endl<<setw(30)<<"Enter a Character to Skip Options Screen.\n";

```

```

if (ran){
    cout<<"You've played the game already!\n";

```

```

}

cin>>slctn; // selection

cout<<endl;

do{ // loops the switch case for user to go through as many times.

    switch(slctn){

        case '1': // basic introduction

            cout<<endl<<"A half a stack of cards, only the suit of Hearts and Spades will be shuffled (excluding
Joker and Royals). \n"

            <<"20 Cards (Ace-9) will be laid out individually and the goal is to match two cards in the least
amount of turns.. \n"

            <<"A matching pair is defined as two cards with the same rank. (two Aces, two fives, etc.) \n"

            <<"If the two cards make a pair, you take them and count that as a point. This is a memory game!\n"

            <<"Project 1 will have you try to find the matching card for letter a, instead of all 20.\n"

            <<"Since there isn't a visual stack of cards, the user must type a placement (exa. a b, f a), after
choosing, the cards will be revealed.\n"<<endl;

            cout<<endl<<"Would you like to select another option? (type 1,2,3/ if not any type any other
character.)\n";

            cin>>slctn;

            break;

        case '2': // scores

            if (score>=0){ // still in update for project 2

                if (score==0){ // starting score

                    cout<<fixed<<setprecision(2);

                    cout<<endl<<"Your score is currently "<<score<<".\n";

                    cout<<"Proj 1, V.1 WIP (more added in project 2)\n";

                }

                else{ // after run through score

                    cout<<fixed<<setprecision(2);

                    cout<<endl<<"Your score is currently "<<score<<".\n";

                    cout<<"Proj 1, V.1 WIP (more added in project 2)\n";

                }

                cout<<endl<<"Would you like to select another option? (type 1,2,3/ if not any type any other
character.)\n";

                cin>>slctn; break;

```



```

    }
    case '3': // average
        cout<<fixed<<setprecision(2)<<showpoint; // iomanip library use
        avg = 0.05 *static_cast<float>(score); // 0.05 from 1/20
        cout<<endl<<"Your average guessing rate is currently "<<avg<<endl
            <<"Rounded, it is "<<round(avg)<<"!\n"; // utilizing round from cmath library
        cout<<"Proj 1, V.1 WIP (more added in project 2)\n";
        cout<<endl<<"Would you like to select another option? (type 1,2,3/ if not any type any other
character.)\n";
        cin>>slctn;
        break;
        default:break; //ends case '3'
    }
}while(slctn=='1' || slctn=='2' || slctn=='3'); // selection if equals 1,2,or 3.
// If the user wants to play the game
cout<<"Want To Play?(y/any other character) \n"<<endl; // user input to begin game
cin>>chce1;
cout<<endl<<endl<<endl;

chce1+=chce1<=97?32:0; // uppercase to lowercase
chce2="";
while(chce1 == 'y' && chce2 != "over"){
    cout<<setw(34)<<"The Concentration Card Board!\n";
    cout<<setw(24)<<"_____ \n";
    cout<<setw(25)<<"| a b c d | \n";
    cout<<setw(25)<<"| e f g h | \n";
    cout<<setw(25)<<"| i j k l | \n";
    cout<<setw(25)<<"| m n o p | \n";
    cout<<setw(25)<<"| q r s t | \n";
    cout<<setw(24)<<"_____ \n";
    char a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t, userG1, userG2; // for comparison from string letter to char
    a=card1; //takes value from initial card1 (etc)

```

```

b=card2;
c=card3;
d=card4;
e=card5;
f=card6;
g=card7;
h=card8;
i=card9;
j=card10;//takes value from initial card10 (etc)
k=card11;
l=card12;
m=card13;
n=card14;
o=card15;
p=card16;
q=card17;
r=card18;
s=card19;
t=card20;//takes value from initial card20 (etc)
cout<<"This is your board, try to find the matching pair to a! \n"
    "Type two letters to see if you found a match! (exa. a g)\n";

unsigned int moves = 0; // counter for amount of moves made
bool complete = true; // initializes
while(complete && chce2 != "over"){ //stops when pair is found
    cin>>userG1>>userG2; // users guess 1 and 2

    // user guesses a b, b a
    if ((userG1=='a' && userG2=='b' && ((a-10)==b || (a+10)==b)) || (userG1=='b' && userG2=='a' && ((b-10)==a || (b+10)==a))){cout<<"That was a pair!\n";

        cout<<"You guessed the "<<newCd1<<" and "<<newCd2<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

```

```

// user guesses a c, c a

else if ((userG1=='a' && userG2=='c' && ((a-10)==c || (a+10)==c)) || (userG1=='c' && userG2=='a' && ((c-10)==a || (c+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd3<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a d, d a

else if ((userG1=='a' && userG2=='d' && ((a-10)==d || (a+10)==d)) || (userG1=='d' && userG2=='a' && ((d-10)==a || (d+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd4<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a e, e a

else if ((userG1=='a' && userG2=='e' && ((a-10)==e || (a+10)==e)) || (userG1=='e' && userG2=='a' && ((e-10)==a || (e+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd5<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a f, f a

else if ((userG1=='a' && userG2=='f' && ((a-10)==f || (a+10)==f)) || (userG1=='f' && userG2=='a' && ((f-10)==a || (f+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd6<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a g, g a

else if ((userG1=='a' && userG2=='g' && ((a-10)==g || (a+10)==g)) || (userG1=='g' && userG2=='a' && ((g-10)==a || (g+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd7<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a h, h a

else if ((userG1=='a' && userG2=='h' && ((a-10)==h || (a+10)==h)) || (userG1=='h' && userG2=='a' && ((h-10)==a || (h+10)==a))){cout<<"That was a pair!\n";

    cout<<"You guessed the "<<newCd1<<" and "<<newCd8<<".\n"; cout<<"It took you "<<moves+1<<"
move(s) to find the pair! \n";complete=false;score=moves+1;}

// user guesses a i, i a

```

```
else if ((userG1=='a' && userG2=='i' && ((a-10)==i || (a+10)==i)) || (userG1=='i' && userG2=='a' && ((i-10)==a || (i+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd9<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a j, j a
```

```
else if ((userG1=='a' && userG2=='j' && ((a-10)==j || (a+10)==j)) || (userG1=='j' && userG2=='a' && ((j-10)==a || (j+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd10<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a k, k a
```

```
else if ((userG1=='a' && userG2=='k' && ((a-10)==k || (a+10)==k)) || (userG1=='k' && userG2=='a' && ((k-10)==a || (k+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd11<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a l, l a
```

```
else if ((userG1=='a' && userG2=='l' && ((a-10)==l || (a+10)==l)) || (userG1=='l' && userG2=='a' && ((l-10)==a || (l+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd12<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a m, m a
```

```
else if ((userG1=='a' && userG2=='m' && ((a-10)==m || (a+10)==m)) || (userG1=='m' && userG2=='a' && ((m-10)==a || (m+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd13<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a n, n a
```

```
else if ((userG1=='a' && userG2=='n' && ((a-10)==n || (a+10)==n)) || (userG1=='n' && userG2=='a' && ((n-10)==a || (n+10)==a))) {cout<<"That was a pair!\n";
```

```
    cout<<"You guessed the "<<newCd1<<" and "<<newCd14<<".\n"; cout<<"It took you "<<moves+1<<"  
move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a o, o a
```

```
    else if ((userG1=='a' && userG2=='o' && ((a-10)==o || (a+10)==o)) || (userG1=='o' && userG2=='a' && ((o-10)==a || (o+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd15<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a p, p a
```

```
    else if ((userG1=='a' && userG2=='p' && ((a-10)==p || (a+10)==p)) || (userG1=='p' && userG2=='a' && ((p-10)==a || (p+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd16<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a q, q a
```

```
    else if ((userG1=='a' && userG2=='q' && ((a-10)==q || (a+10)==q)) || (userG1=='q' && userG2=='a' && ((q-10)==a || (q+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd17<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a r, r a
```

```
    else if ((userG1=='a' && userG2=='r' && ((a-10)==r || (a+10)==r)) || (userG1=='r' && userG2=='a' && ((r-10)==a || (r+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd18<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a s, s a
```

```
    else if ((userG1=='a' && userG2=='s' && ((a-10)==s || (a+10)==s)) || (userG1=='s' && userG2=='a' && ((s-10)==a || (s+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd19<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// user guesses a t, t a
```

```
    else if ((userG1=='a' && userG2=='t' && ((a-10)==t || (a+10)==t)) || (userG1=='t' && userG2=='a' && ((t-10)==a || (t+10)==a))){cout<<"That was a pair!\n";
```

```
        cout<<"You guessed the "<<newCd1<<" and "<<newCd20<<".\n"; cout<<"It took you "<<moves+1<<" move(s) to find the pair! \n";complete=false;score=moves+1;}
```

```
// incorrect user guesses a b, b a
```

```
else if ((userG1=='a' && userG2=='b' && ((a-10)!=b || (a+10)!=b)) || (userG1=='b' && userG2=='a' && ((b-10)!=a || (b+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='b') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd2<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd2<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a c, c a
```

```
else if ((userG1=='a' && userG2=='c' && ((a-10)!=c || (a+10)!=c)) || (userG1=='c' && userG2=='a' && ((c-10)!=a || (c+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='c') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd3<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd3<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a d, d a
```

```
else if ((userG1=='a' && userG2=='d' && ((a-10)!=d || (a+10)!=d)) || (userG1=='d' && userG2=='a' && ((d-10)!=a || (d+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='d') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd4<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd4<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a e, e a
```

```
else if ((userG1=='a' && userG2=='e' && ((a-10)!=e || (a+10)!=e)) || (userG1=='e' && userG2=='a' && ((e-10)!=a || (e+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='e') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd5<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd5<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a f, f a
```

```
else if ((userG1=='a' && userG2=='f' && ((a-10)!=f || (a+10)!=f)) || (userG1=='f' && userG2=='a' && ((f-10)!=a || (f+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='f') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd6<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd6<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a g, g a
```

```
else if ((userG1=='a' && userG2=='g' && ((a-10)!=g || (a+10)!=g)) || (userG1=='g' && userG2=='a' && ((g-10)!=a || (g+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='g') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd7<<".\n";}
```

```
else {cout<<"You guessed the "<<newCd7<<" and the "<<newCd1<<".\n";} moves+=1;}
```

```
// incorrect user guesses a h, h a
```

```

        else if ((userG1=='a' && userG2=='h' && ((a-10)!=h || (a+10)!=h)) || (userG1=='h' && userG2=='a' && ((h-10)!=a || (h+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='h'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd8<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd8<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a i, i a

```

```

        else if ((userG1=='a' && userG2=='i' && ((a-10)!=i || (a+10)!=i)) || (userG1=='i' && userG2=='a' && ((i-10)!=a || (i+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='i'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd9<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd9<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a j, j a

```

```

        else if ((userG1=='a' && userG2=='j' && ((a-10)!=j || (a+10)!=j)) || (userG1=='j' && userG2=='a' && ((j-10)!=a || (j+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='j'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd10<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd10<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a k, k a

```

```

        else if ((userG1=='a' && userG2=='k' && ((a-10)!=k || (a+10)!=k)) || (userG1=='k' && userG2=='a' && ((k-10)!=a || (k+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='k'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd11<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd11<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a l, l a

```

```

        else if ((userG1=='a' && userG2=='l' && ((a-10)!=l || (a+10)!=l)) || (userG1=='l' && userG2=='a' && ((l-10)!=a || (l+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='l'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd12<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd12<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a m, m a

```

```

        else if ((userG1=='a' && userG2=='m' && ((a-10)!=m || (a+10)!=m)) || (userG1=='m' && userG2=='a' && ((m-10)!=a || (m+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='m'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd13<<".\n";}

```

```

        else {cout<<"You guessed the "<<newCd13<<" and the "<<newCd1<<".\n";} moves+=1;}

```

```

// incorrect user guesses a n, n a

```

```

        else if ((userG1=='a' && userG2=='n' && ((a-10)!=n || (a+10)!=n)) || (userG1=='n' && userG2=='a' && ((n-10)!=a || (n+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='n'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd14<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd14<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a o, o a

```

```

        else if ((userG1=='a' && userG2=='o' && ((a-10)!=o || (a+10)!=o)) || (userG1=='o' && userG2=='a' && ((o-10)!=a || (o+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='o'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd15<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd15<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a p, p a

```

```

        else if ((userG1=='a' && userG2=='p' && ((a-10)!=p || (a+10)!=p)) || (userG1=='p' && userG2=='a' && ((p-10)!=a || (p+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='p'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd16<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd16<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a q, q a

```

```

        else if ((userG1=='a' && userG2=='q' && ((a-10)!=q || (a+10)!=q)) || (userG1=='q' && userG2=='a' && ((q-10)!=a || (q+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='q'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd17<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd17<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a r, r a

```

```

        else if ((userG1=='a' && userG2=='r' && ((a-10)!=r || (a+10)!=r)) || (userG1=='r' && userG2=='a' && ((r-10)!=a || (r+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='r'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd18<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd18<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a s, s a

```

```

        else if ((userG1=='a' && userG2=='s' && ((a-10)!=s || (a+10)!=s)) || (userG1=='s' && userG2=='a' && ((s-10)!=a || (s+10)!=a))){cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='s'){cout<<"You guessed the "<<newCd1<<" and the "<<newCd19<<".\n";}

```

```

        else{cout<<"You guessed the "<<newCd19<<" and the "<<newCd1<<".\n";}moves+=1;}

```

```

// incorrect user guesses a t, t a

```



```

        else if ((userG1=='a' && userG2=='t' && ((a-10)!=t || (a+10)!=t)) || (userG1=='t' && userG2=='a' && ((t-10)!=a || (t+10)!=a))) {cout<<"Not a pair.\n"; if(userG1=='a' && userG2=='t') {cout<<"You guessed the "<<newCd1<<" and the "<<newCd20<<".\n";}

        else {cout<<"You guessed the "<<newCd20<<" and the "<<newCd1<<".\n";} moves+=1;}

    }

    cout<<endl<<"If you'd like to end the game, type over, if you want to keep trying type anything else. \n";

    cout<<"Reminder, most, if not each time you play the game, it'll be different! (Just re-run the game!)\n";

    cin>>chce2; // ends loop of game, can get better score if tried again.

}

    cout<<"Are you completely done? If you'd like to go back to the home screen type y, if not, any other character will do. ";

    ran=true; // if user has run the game

    cin>>gameSes; // ends program if any other character then y typed.

}

//Exit the program

return 0;

}

```