```cpp
/*
 * shuffle.cpp
 *
 *  Created on: Jan 30, 2017
 *      Author: Brian Clinkenbeard
 */

#include "header.h"
#include "card.h"

/* 1b. perform a perfect shuffle */
void shuffleDeck(card shuffle[52])
{
        /* preserve original deck (cannot copy array) */
        card unshuffled[52];
        for (int i = 0; i < 52; ++i) {
                unshuffled[i] = shuffle[i];
        }

        /* perfect shuffle */
        int firstHalf = 0;
        int secondHalf = 26;
        int count = 0;
        while (count < 51) {
                shuffle[count] = unshuffled[firstHalf];
                firstHalf++;
                count++;
                shuffle[count] = unshuffled[secondHalf];
                secondHalf++;
                count++;
        }
}


```

```
/*
 * main.cpp
 *
 *  Created on: Jan 30, 2017
 *      Author: Brian Clinkenbeard
 */

#include "header.h"
#include "card.h"

int main()
{
        /* 1b. initialize deck */
        card deck[52];

        /* iterate through the enums */
        int count = 0;
        for (int i = 0; i < 4; ++i) {
                suit initSuit = static_cast<suit>(i);
                for (int j = 0; j < 13; ++j) {
                        face initFace = static_cast<face>(j);
                        deck[count] = card(initFace, initSuit);
                        count++;
                }
        }

        /* perfect shuffle */
        card shuffled[52] = deck;
        shuffleDeck(shuffled);


        /* shuffle until deck is returned to original */
        card final[52] = shuffled;
        count = 1; /* one shuffle has already occurred */
        while (!equalDecks(deck, final)) {
                shuffleDeck(final);
                count++;
        }

        /* 1c. print decks */
        cout << "Initial deck:" << endl;
        printDeck(deck);
        cout << "Shuffled deck:" << endl;
        printDeck(shuffled);
        cout << "Final deck:" << endl;
        printDeck(final);

        /* 1d. print amount of shuffles */
        cout << "It took " << count << " perfect shuffles for the deck to return to its original order." << endl;

        return 0;
}
```

```
/*
 * card.cpp
 *
 *  Created on: Jan 30, 2017
 *      Author: Brian Clinkenbeard
 */

#include "header.h"
#include "card.h"

/* for array instantiation */
card::card() {}

/* constructor */
card::card(face newFace, suit newSuit)
{
        myFace = newFace;
        mySuit = newSuit;
}

/* "break;" not necessary on returns */
string card::getFace()
{
        switch (myFace) {
        case ACE:
                return "Ace";
        case TWO:
                return "2";
        case THREE:
                return "3";
        case FOUR:
                return "4";
        case FIVE:
                return "5";
        case SIX:
                return "6";
        case SEVEN:
                return "7";
        case EIGHT:
                return "8";
        case NINE:
                return "9";
        case TEN:
                return "10";
        case JACK:
                return "Jack";
        case QUEEN:
                return "Queen";
        case KING:
                return "King";
        }
}

string card::getSuit()
{
        switch (mySuit) {
        case CLUBS:
                return "Clubs";
        case DIAMONDS:
                return "Diamonds";
        case HEARTS:
                return "Hearts";
        case SPADES:
                return "Spades";
        }
}

/* print the face and suit of the card */
ostream& operator<<(ostream &os, card &outcard)
{
        os << outcard.getFace() << " of " << outcard.getSuit();
        return os;
}

/* overload relational operators for comparison */
bool operator==(const card &firstCard, const card &secondCard)
{
        return (firstCard.myFace == secondCard.myFace && firstCard.mySuit == secondCard.mySuit);
}

bool operator!=(const card &firstCard, const card &secondCard) {
        return !(firstCard == secondCard);
}
```

```
/*
 * print.cpp
 *
 *  Created on: Jan 30, 2017
 *      Author: Brian Clinkenbeard
 */

#include "header.h"
#include "card.h"

/* 1b. print the deck of cards */
void printDeck(card printDeck[52])
{
        for (int i = 0; i < 52; ++i) {
                cout << i + 1 << ". " << printDeck[i] << endl;
        }
}
```

```cpp
/*
 * compare.cpp
 *
 *  Created on: Jan 30, 2017
 *      Author: Brian Clinkenbeard
 */

#include "header.h"
#include "card.h"

/* 1b. compare two decks of cards */
bool equalDecks(const card firstDeck[52], const card secondDeck[52])
{
        for (int i = 0; i < 52; ++i) {
                if (firstDeck[i] != secondDeck[i])
                        return false;
        }
        /* fall-through: all cards are equal */
        return true;
}
```