0. Command-Line Parameter

The user can provide an optional integer argument for the random number generator seed when starting the program:

```
./straights 44
```

This allows for pseudo-random shuffling. The provided \texttt{shuffle} algorithm declares a \texttt{static} random number generator that is used to randomly shuffle cards in the deck. The random number generator is seeded with the global variable \texttt{seed}\(^1\), which is:

1) defined in the same .\texttt{cc} file as \texttt{shuffle},
2) declared in the corresponding .\texttt{h} file, and
3) set at the start\(^2\) of the \texttt{main()} program to be the integer value of the provided command-line argument if one is provided; otherwise, the default seed value is used.

Games that are started with the same seed value have the same sequences of deals.

1. Invite Players

At the beginning of the program, prompt the user with the following message:

```
Is player <x> a human(h) or a computer(c)?
```

where \(<x>\) is the id number representing the player being initialized. The user then types either 'h' or 'c' to set the desired player type. Repeat this step for each of the four players.

2. Shuffling and Dealing

Initially, the cards in the deck must be in the following order\(^3\):

```
AC 2C 3C ... TC JC QC KC AD 2D ... QD KD AH 2H ... QH KH AS 2S ... QS KS
```

At the beginning of every round, shuffle the deck once (using the provided \texttt{shuffle} function in your solution). Do not shuffle the cards in any other way. After the shuffle, deal out the cards such that the first 13 cards belong to Player 1, the next 13 cards belong to Player 2, the next 13 belong to Player 3, and the last 13 cards belong to Player 4.

When the round is over, the deck is restored to its previous state before shuffling it. (If it was reset to the original state, then the players would end up with the exact same hands of cards again.) For example, if after shuffling the deck has the order 6S 2S 4C 2H 6C AD JD 8D 2D AS JH 7S AH 2C 3H 3S KD 5C 5H 9C 3D JC 6H AC TH 4D 5D TD 7D 4S 7H 4H TC 9D JS KC 8S KS TS QC 9H 7C 9S QS 5S 8C 8H KH 6D QD QH 3C, then before it is shuffled for the next round, the cards are

\(^1\) This is the only global variable you are allowed to have in your program.
\(^2\) It must only be set once; otherwise, the output will not match.
\(^3\) Note that the 10 is represented by the letter 'T'. This way, all card ranks can be represented by a single character.
Specifications

5C 5H 9C 3D 3C 6H AC TH 4D 5D TD 7D 4S 7H 4H TC 9D JS 8S TS QC 9H 7C 9S QS 5S 8C 8H KH 6D QD QH 3C. This way, the results will be repeatable.

3. Gameplay—Start

The game starts after the shuffle and the deal. The four players take turns to play their cards. First, print the following line (regardless of whether the first player is a human):

A new round begins. It’s player <x>’s turn to play.

where <x> depends on who has the 7 of spades.

4. Gameplay—Human Player

Whenever it is a human player’s turn, print the following 8 lines:

Cards on the table:
Clubs:<list of clubs>
Diamonds:<list of diamonds>
Hearts:<list of hearts>
Spades:<list of spades>
Your hand:<cards in your hand>
Legal plays:<legal plays in your hand>

Each of <list of spades>, <list of hearts>, <list of clubs>, and <list of diamonds> is an ordered sequence of all the ranks in that suit (e.g., 6 7 8 9 T J Q) that have already been played.

<cards in your hand> and <legal plays in your hand> are lists of cards in the player’s hand, where each card is in the form <rank><suit> (e.g. 7C). Print the cards in the same order that they appear in the deck. Do not rearrange the cards. Every list of cards, except for that produced by the deck command, starts with a space character, and a single space separates each card from the next. There is no space after the final card in the list. If there are no legal plays, then the list of cards consists of an empty string i.e. print Legal plays:

The program then waits for the user to enter a command.

5. Gameplay—Commands

There are 5 valid commands in this game:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| play <card>   | Play the specified card. You may assume that the <card> has valid syntax (i.e., <rank><suit>, such as 7C), and that the specified card is in the player’s hand. However, it might not be a legal play. If the play is legal, print: Player <x> plays <card>.


and proceed to the next player. Otherwise, print:

This is not a legal play.

and do not proceed to the next player until a legal play is made.

discard <card>

If the player has no legal plays, discard the specified card from the player’s hand into the player’s discard pile. Again, assume that the <card> has valid syntax and that the <card> is in the player’s hand. For test purposes, the value of the card is printed even though that deviates from the normal straights game play.

Player <x> discards <card>.

Otherwise, print the following error message:

You have a legal play. You may not discard.

deck

Print the contents of the deck in order, 13 cards per line. For example:

TS 2D 3S KH 3H 2C 5D TC 8S TD AC KC QH
4D 3H 6H JC KD 8C 7D TH 4H 9S 6S 4S KS
7S 7C QD 6C 2H 6D 3C 9C 5H 3D AD 5S 8H
QC 2S 8D JS QS AS JD 4C 7H 9D 5C AH 9H

(This command is not part of the straights game. It is provided to help both you and us with the testing and debugging of your program.)

quit

Terminate the program immediately.

ragequit

Filled with anger, a human player decides to leave! Print the following message:

Player <x> ragequits. A computer will now take over.

Replace the current human player with a computer player, and resume the game.

6. Gameplay—Computer Player

If it is a computer player’s turn to play, print either one of the two lines, as appropriate:

Player <x> plays <card>.

or
Player <x> discards <card>.

Proceed to the next player.

The computer player that you will implement is very simple. It always makes the first legal play in its hand. If there are no legal plays, the first card in its hand is discarded.

7. Scoring
When all of the cards have been played, the round ends. At this point, print the following lines for each of the players:

Player <x>’s discards:<list of discards>
Player <x>’s score: <old score> + <score gained> = <new score>

<List of discards> is a list of the cards the player discarded in the current round. Print the cards in the same order that they were discarded.

If one player has accumulated ≥ 80 points, the game ends. (Note that there is no prompt to replay again. Instead, you would just run the executable again.) The player with the lowest score wins. Print the following victory message for the winning player(s):

Player <x> wins!

If multiple players tie for the win, print the above message for each winner.

If no one exceeded the score limit, then reshuffle the deck and begin another round.