1. Command Line Parameter

To start the program, the user can provide an optional integer argument, like this:

```
./straights 44
```

This allows for pseudo---random shuffling. At the beginning of your program, call the `srand48` function (see `man srand48`) with the provided argument. Games that are started with the same argument value have the same sequences of deals.

2. 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)?
>
```

The user will then type either "h" or "c". Repeat this step for each of the four players.

3. Shuffling and Dealing

Initially, the cards in the deck should be in the following order:

```
AC 2C 3C ... 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 shuffle function in your solution). Do not shuffle the cards in any other way. After the shuffle, assume 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.

4. Gameplay — Start

The game starts after the shuffle and the deal. The four players will 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.

5. 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>, <list of diamonds> is an ordered sequence of all the ranks in that suit (e.g., `6 7 8 9 10 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. `7S`). Print the cards in the same order that they appear in the deck. Do not rearrange the cards.

The program will then wait for the user to enter a command.

6. Gameplay — Commands
There are 5 valid commands in this game:

```
play <card>
discard <card>
deck
quit
ragequit
```

a) 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.

b) discard <card>
Discard the specified card. Again, assume that the <card> has valid syntax and that the <card> is in the player's hand.

If the player has no legal plays, the card will be discarded.

```
Player <x> discards <card>.
```

Otherwise, print the following error message.

```
You have a legal play. You may not discard.
>
```

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

```
10S 2D 3S KH 3H 2C 5D 10C 8S 10D AC KC QH
4D JH 6H JC KD 8C 7D 10H 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.)

d) quit
Terminate the program immediately.

e) 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:

```
Player <x> plays <card>.
```

or

```
Player <x> discards <card>.
```

and 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 will be 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 or more, the game ends. The player with the lowest score wins. Print the following victory message for the top player.

```
Player <x> wins!
```

If multiple top players tie for first place, print the above message for each winner.

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