Demo Guidelines

Please note the following:

1. Valid commands for the program are as follows:
   `h / c / deck / quit / play <card> / discard <card> / ragequit`
2. Valid card syntax is of the form `<rank><suit>` (e.g. `7S / KD / TH`).
3. This program does not handle invalid input syntax.
4. Please be careful when entering commands and cards.

Building the program, straights:

Issuing the command, `make` will compile and build the executable named straights. This executable implements all the requirements of the project specifications. The result is as follows:

```
saabaksh@ubuntu2004-008:~/cs246/f21/project$ make
g++ -std=c++14 -MMD   -c -o card.o card.cc
g++ -std=c++14 -MMD   -c -o deck.o deck.cc
g++ -std=c++14 -MMD   -c -o table.o table.cc
g++ -std=c++14 -MMD   -c -o player.o player.cc
g++ -std=c++14 -MMD   -c -o computer.o computer.cc
g++ -std=c++14 -MMD   -c -o human.o human.cc
g++ -std=c++14 -MMD   -c -o game.o game.cc
g++ -std=c++14 -MMD   -c -o main.o main.cc
g++ card.o deck.o table.o player.o computer.o human.o game.o main.o -o straights
saabaksh@ubuntu2004-008:~/cs246/f21/project$
```

Command line parameters:

The executable, namely straights may be provided with an optional integer argument *seed* as follows:

```
saabaksh@ubuntu2004-002:~/cs246/f21/project$ ./straights 13
```

If no *seed* is provided, the current system time is used as the default *seed*. In the following examples, we will make use of the *seed* 13 as shown in the example above.

Example 1:

Issue the command, `./straights 13`. The following message is printed:

```
Is Player1 a human (h) or a computer (c)?
>
```

Note the symbol "> " prompts the user for input. We may enter `h` or `c` to initialize a human or computer player respectively. In this example, players 2 and 3 are humans and players 1 and 4 are computers.

The players are initialized as follows:

```
Is Player1 a human (h) or a computer (c)?
> c
Is Player2 a human (h) or a computer (c)?
> h
Is Player3 a human (h) or a computer (c)?
> h
Is Player4 a human (h) or a computer (c)?
> c
```

Right after initialization, the following message is printed:

```
A new round begins. It's Player2's turn to play.
```

Note that this message is printed at the start of every round and the player's turn is decided based on who has the card `7S`. In this instance, player 2 has it. At the start of each player's turn, regardless of whether human or computer, the cards on the table, current player's hand and legal plays are printed in the following format:

```
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades:
Your hand: AD 4D 6H QH 2S KS 3C 3H 3S 7D 9S 8D 7S
Legal plays: 7S
>
```

Before moving on, assume human players will only ever try to play or discard a card in their current hand as per the specifications of the project.

Note that at this point, player 2 can only play `7S`. Issue the following commands:

1. `play 7D`
2. `play 3S`
3. `discard 7S`
4. `discard 4D`

The following output is printed:

```
> play 7D
This is not a legal play.
> play 3S
This is not a legal play.
> discard 7S
You have a legal play. You may not discard.
> discard 4D
You have a legal play. You may not discard.
>
```

Note for commands 1 and 2, the output message states that the play is not legal since `7D` and `3S` are not in the list of legal plays. Additionally, commands 3 and 4 are invalid because discarding is only allowed when the legal plays are empty which is not the case in this instance.

Now, issue the command `play 7S`.

```
> play  7S
Player2 plays 7S.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 7
Your hand: 2D 5H QD 6S QC 8H KC KH 2H AS JH 4C 9D
Legal plays: 6S
>
```

As expected, the program accepts the only legal play, prints the affirmation message, i.e., "`Player2 plays 7S.`" and prints the table for player 3.

Next, issue the command `play 6S`. The following is printed:

```
> play 6S
Player3 plays 6S.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 6 7
Your hand: 5S TD 7C 5D QS 5C AC 4H 7H 6C TS 4S TH
Legal plays: 5S 7C 7H
Player4 plays 5S.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 5 6 7
Your hand: JS KD 6D 9H 8S 8C 3D JD 9C JC 2C TC AH
Legal plays: 8S
Player1 plays 8S.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 5 6 7 8
Your hand: AD 4D 6H QH 2S KS 3C 3H 3S 7D 9S 8D
Legal plays: 7D 9S
>
```

Once again, the expected takes place. However, notice the behavior of the computer players 1 and 4. Player 4 has three cards it can play, namely `5S`, `7C`, `7H` but it plays `5S`. This is because it was the first card in its list of legal plays. Player 1 plays `8S` as it is the only available play.

Next, play the second card in the list of legal plays, i.e., issue the command `play 9S`.

```
> play 9S
Player2 plays 9S.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 5 6 7 8 9
Your hand: 2D 5H QD QC 8H KC KH 2H AS JH 4C 9D
Legal plays:
>
```

The expected takes place even though human player 2 played the second card in the list.

Note that at this point, player 3 has no legal plays and can only discard from its current hand. Issue the following commands:

1. `play 2D`
2. `play JH`

The following output is printed:

```
> play 2D
This is not a legal play.
> play JH
This is not a legal play.
>
```

Note for both commands, the output message states that the play is not legal since `2D` and `JH` are not in the empty list of legal plays.

Before we move on, issue the command `deck`. The following is printed:

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

Note that this works as intended by the project specifications.

Now, issue the command `discard KC`.

```
> discard KC
Player3 discards KC.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 5 6 7 8 9
Your hand: TD 7C 5D QS 5C AC 4H 7H 6C TS 4S TH
Legal plays: 7C 7H TS 4S
Player4 plays 7C.
Cards on the table:
Clubs: 7
Diamonds:
Hearts:
Spades: 5 6 7 8 9
Your hand: JS KD 6D 9H 8C 3D JD 9C JC 2C TC AH
Legal plays: 8C
Player1 plays 8C.
Cards on the table:
Clubs: 7 8
Diamonds:
Hearts:
Spades: 5 6 7 8 9
Your hand: AD 4D 6H QH 2S KS 3C 3H 3S 7D 8D
Legal plays: 7D
>
```

As expected, the program accepts the discard, prints the affirmation message, i.e., "`Player3 discards KC.`" and prints the next computer players' behaviors up until the next human player.

Now, issue the command `ragequit`. The following is printed:

```
> ragequit
Player2 ragequits. A computer will now take over.
Player2 plays 7D.
Cards on the table:
Clubs: 7 8
Diamonds: 7
Hearts:
Spades: 5 6 7 8 9
Your hand: 2D 5H QD QC 8H KH 2H AS JH 4C 9D
Legal plays:
>
```

As expected, the program prints the affirmation message, i.e., "`Player2 ragequits. A computer will now take over.`" and player 2 plays its turn as a computer.

Now, issue the command `quit`. The program is terminated immediately as shown below.

```
> quit
saabaksh@ubuntu2004-008:~/cs246/f21/project$
```

Example 2:

The following example is seeded with 13 and all its players are initialized as computers as shown below:

```
saabaksh@ubuntu2004-002:~/cs246/f21/project$ ./straights 13
Is Player1 a human (h) or a computer (c)?
> c
Is Player2 a human (h) or a computer (c)?
> c
Is Player3 a human (h) or a computer (c)?
> c
Is Player4 a human (h) or a computer (c)?
> c
```

Three instances of example 2 are highlighted below.

i. Instance 1:

```
Cards on the table:
Clubs:
Diamonds: 7
Hearts:
Spades: 5 6 7 8
Your hand: 2D 5H QD QC 8H KC KH 2H AS JH 4C 9D
Legal plays:
Player3 discards 2D.
```

It is player 3's second turn in round 1. The computer player can discard any card from its hand but chooses to discard `2D`. This is because `2D` is the first card in its hand.

ii. Instance 2:

```
Player1 discards AH.
Player1's discards: KD AH
Player1's score: 0 + 14 = 14
Player2's discards: AD QH 2S 3H
Player2's score: 0 + 18 = 18
Player3's discards: 2D 5H QD QC KC KH 2H AS
Player3's score: 0 + 60 = 60
Player4's discards: 4H
Player4's score: 0 + 4 = 4
A new round begins. It's Player1's turn to play.
```

The first round has come to an end. However, none of the players' scores have surpassed the maximum score, i.e., 80. As a result, a new round begins.

iii. Instance 3:

```
Player1's discards:
Player1's score: 39 + 0 = 39
Player2's discards: AD KH AS KC
Player2's score: 19 + 28 = 47
Player3's discards: QH TC JC 3S QC
Player3's score: 64 + 48 = 112
Player4's discards: 4S 2S
Player4's score: 33 + 6 = 39
Player1 wins!
Player4 wins!
```

The third round comes to an end. This time, however, players 3's score has surpassed the maximum score. As a result, the players with the lowest score win. In this case, players 1 and 4 tie for a draw as they both have the lowest score of 39. And so, the program prints the winner message for both players.

Example 3:

The following example is once again seeded with 13 and all its players are initialized as computers except player 2 as shown below:

```
Is Player1 a human (h) or a computer (c)?
[> c
Is Player2 a human (h) or a computer (c)?
[> h
Is Player3 a human (h) or a computer (c)?
[> c
Is Player4 a human (h) or a computer (c)?
[> c
```

Two instances of example 3 are highlighted below.

i. Instance 1:

It is player 2's first turn in round 1. Issue the command `deck`. The following is printed:

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

Note that to get to instance 2, act as a computer and play the first card in the list of legal plays. If there are no legal plays, discard the first card in player 2's hand. Stop when a new round begins.

ii. Instance 2:

It is player 2's first turn in round 2. Issue the command `deck`. The following is printed:

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

These instances collectively prove the reshuffling of the deck at the beginning of each round.