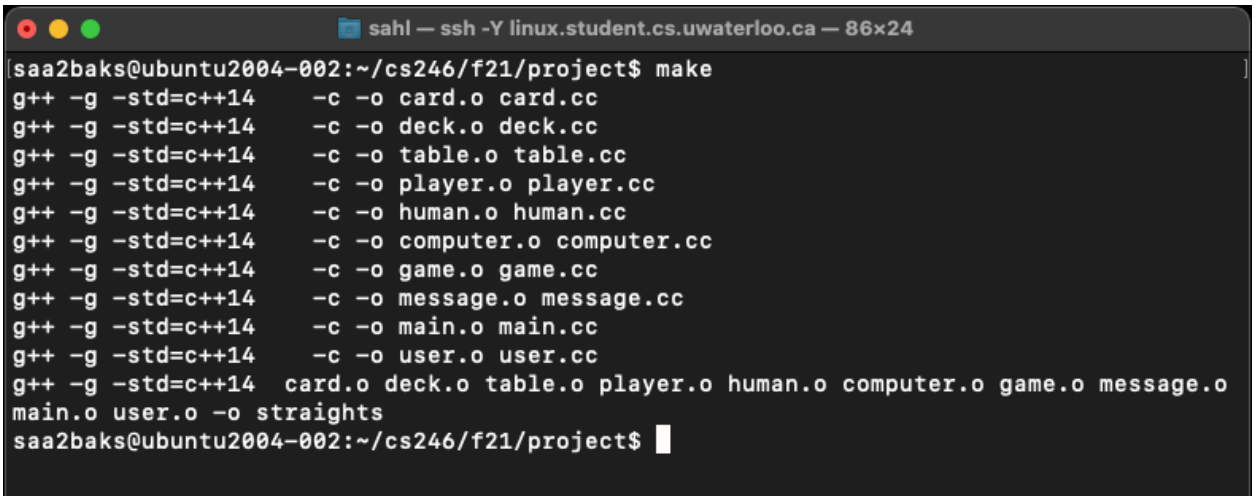
Demo

Note

Please be careful when entering commands or cards.
This program is unable to handle invalid input syntax.
The valid commands are h/c/play/discard/quit/ragequit/deck.
The valid Card syntax is <rank><suit>. For example, 7S/AC/JH.

Start

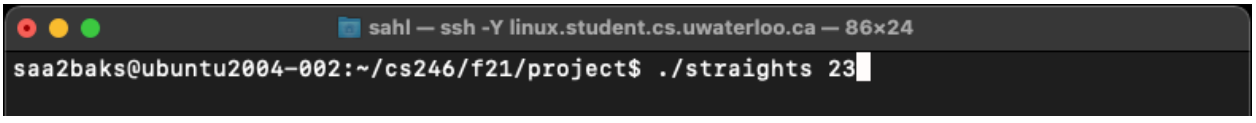
You can compile the program using the make command.



After compiling, the executable will be named “straights”.
This executable implements all the requirements of the straights.pdf project.

Example 1

To run the program with a specific seed, please enter an optional integer command line argument. In the following example, the seed used is 23.



After this, a prompt will appear to initialize a human or computer player.
Enter h for human or c for computer.

In our example, we will be initializing 2 players to be human and 2 to be computer.

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

Whenever a new round begins. The following message is displayed.
It is Player3's turn to play since this player has the 7 of Spades in their hand.

```
A new round begins. It's Player3's turn to play.
```

Whenever it is a players turn to play, the table, and the current players hand and legal plays is printed as specified in the straights.pdf.

```
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades:
Your hand: TD 7S 3S 6D JH JS 9C 2H 4S 3C 9D 5D 9S
Legal plays: 7S
>
```

At this point, a human player can either play a legal or illegal card using by entering play <card>, discard a legal or illegal card by entering discard <card>, print the contents of the deck by entering deck, terminate the program by entering quit or replace themselves with a computer player by entering ragequit to the prompt >.

However, a computer player will automatically play its first legal play. If no legal plays are possible, it will discard the first card in its hand.

Please ensure that the card played or discarded is in the human players hand as this is assumed in the project specification. It may still be an illegal play or discard.

Let us assume the following happens.

Player3 tries to play an illegal card.
The following output is printed to indicate that the play is not legal.

```
> play TD
This is not a legal play.
>
```

Player3 tries to discard when legal plays are present.
The following output is printed to indicate that a legal play is possible.

```
> discard 2H
You have a legal play. You may not discard.
>
```

Finally, Player3 play the only legal move possible.
The following output is printed to indicate that the play was a success, and the program will now move on to the next player.

```
> play 7S
Player3 plays 7S.
```

It is player4's turn who is a human.

```
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades: 7
Your hand: KD TC 8C 3H 7D 2S JC JD AS 6C AH 9H QC
Legal plays: 7D
>
```

Player4 learnt his lesson from Player3 and directly plays the only legal play possible.

```
> play 7D
Player4 plays 7D.
```

It is player1's turn who is a computer.

```
Cards on the table:
Clubs:
Diamonds: 7
Hearts:
Spades: 7
Your hand: AD KC QD 3D KS 5H QS TH 2C 5S QH AC 5C
Legal plays:
Player1 discards AD.
```

There is no prompt because it is a computer players turn. And because there are no legal plays possible, the computer player will discard the first card in its hand as mentioned previously.

It is player2's turn who is also a computer.

```
Cards on the table:
Clubs:
Diamonds: 7
Hearts:
Spades: 7
Your hand: 8H 7H 7C 4C 8D 8S 4H KH 6H TS 2D 6S 4D
Legal plays: 7H 7C 8D 8S 6S
Player2 plays 7H.
```

Again, there is no prompt. However, this time there are five legal plays possible. Therefore, the computer will play the first legal play as mentioned previously.

It is Player3's turn.

```
Cards on the table:
Clubs:
Diamonds: 7
Hearts: 7
Spades: 7
Your hand: TD 3S 6D JH JS 9C 2H 4S 3C 9D 5D 9S
Legal plays: 6D
>
```

Player3 decides to have a look at the shuffled deck before playing their card.

Player3 does this by entering the deck command.
Player3 then plays their legal play.

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

```
> play 6D
Player3 plays 6D.
```

After Players3's turn, Player 4 receives the following output.
Notice there are no legal plays possible.

```
Cards on the table:
Clubs:
Diamonds: 6 7
Hearts: 7
Spades: 7
Your hand: KD TC 8C 3H 2S JC JD AS 6C AH 9H QC
Legal plays:
>
```

Let us assume the following happens.

Player4 tries to play an illegal card.
The following output is printed to indicate that the play is not legal.

```
> play 2S
This is not a legal play.
```

Player4 discards a card from its hand.
The following output is printed to indicate that the discard was a success, and the program will now move on to the next player.

```
> discard QC
Player4 discards QC.
```

Notice that unlike, a computer player, the human was able to discard any card in their hand.

After playing the game for some time, it is player3's turn and the following output is printed. Notice that there is more than one legal play move available. Also, notice that the players hand has lesser cards as the game progresses.

```
Cards on the table:
Clubs: 4 5 6 7 8
Diamonds: 5 6 7 8 9
Hearts: 7 8 9 T J Q
Spades: 7
Your hand: TD 3S JS 9C 2H 4S 3C 9S
Legal plays: TD 9C 3C
>
```

Let us assume the following:

Player3 plays the second legal play.
The following output is printed to indicate that the play was a success, and the program will now move on to the next player.

```
> play 9C
Player3 plays 9C.
```

Notice, that unlike a computer the human was able to play any legal play of their choice.

It is Player4's turn.

```
Cards on the table:
Clubs: 4 5 6 7 8 9
Diamonds: 5 6 7 8 9
Hearts: 7 8 9 T J Q
Spades: 7
Your hand: KD TC 3H 2S JC JD AS AH
Legal plays: TC
>
```

Player4 is angry the game is taking so long and decides to leave the game and let a computer take over.

Player4 does this by entering the ragequit command.
The following output is printed to indicate that the ragequit command was a success, and a computer will take Player4's turn.

```
> ragequit
Player4 ragequits. A computer will now take over.
Player4 plays TC.
```

Notice that Player4 plays their turn before moving on to the next player.

Player3 decides to stick around until the next round.
Finally, after playing the game for some time, the following output is printed.

```
Cards on the table:
Clubs: 2 3 4 5 6 7 8 9 T J
Diamonds: 4 5 6 7 8 9 T J
Hearts: 3 4 5 6 7 8 9 T J Q K
Spades: 5 6 7 8 9 T J Q
Your hand: 3S
Legal plays:
>
```

Notice that there is only 1 card in the Player3's hand. And since Player 3 was the first player, players 4,2,1 must also have one turn left. Player3 discards the only card in the hand. After which the computer players play their turn, and the first round comes to end.

The following output is printed.

```
Player1's discards: AD KC QD 3D KS QS
Player1's score: 0 + 54 = 54
Player2's discards: 2D
Player2's score: 0 + 2 = 2
Player3's discards: JS 4S 3S
Player3's score: 0 + 18 = 18
Player4's discards: QC KD 2S AS
Player4's score: 0 + 28 = 28
```

After the end of each round, each players discards, and scores are printed and calculated as specified in the straights.pdf project. However, if the none of the players reach a score of 80, a new round will begin, and the scores are carried over.

As mentioned previously, a new round begins, and the following message is displayed.
Notice that it is Player3's turn again. This is because Player3 has the 7 of spades again.

```
A new round begins. It's Player3's turn to play.
```

```
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades:
Your hand: TS JC 8C 8S 9D AH 2C 7S 3H QD 6H 8D AS
Legal plays: 7S
>
```

Let us assume the following:

Player3 enters the deck command.
The following output is printed indicating that the deck was indeed shuffled before the round.

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

Player3 decides he wants to ragequit.
Player 3 enters the ragequit command.

```
> ragequit
Player3 ragequits. A computer will now take over.
Player3 plays 7S.
```

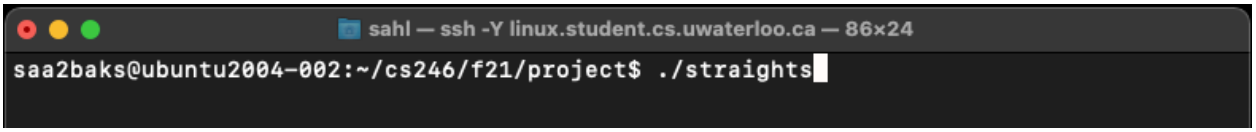
At this point, there are 4 computer players who play the game automatically until any of the player scores reach 80. After which the game will end, and the following output is printed.

```
Player1's discards: QC 3C KC AD
Player1's score: 54 + 29 = 83
Player2's discards: 3D
Player2's score: 2 + 3 = 5
Player3's discards: JC AH 2C
Player3's score: 18 + 14 = 32
Player4's discards: 4D 2D AC
Player4's score: 28 + 7 = 35
Player2 wins!
```

Notice that only one player had the score of 80 but the game still ends. And since Player 2 has the lowest score, Player2 is declared the winner.

Example 2

To run the program without a specific seed, please do *not* enter an optional integer command line argument, as in the following example.



In this example, we initialize all players to be human.

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

It is Player4's tun to play.
Player4 decides to stop the game.

```
A new round begins. It's Player4's turn to play.
Cards on the table:
Clubs:
Diamonds:
Hearts:
Spades:
Your hand: 4C TH 8C 5D JC AS 7S 8D 6D 2S 7C 2D 9S
Legal plays: 7S
>
```

Player4 does this by entering the quit command.
And the program is terminated immediately.

```
> quit
saa2baks@ubuntu2004-002:~/cs246/f21/project$
```

Some tests

Type	Seed used	
Multiple winners with all computer players	13	✅
One winner with all computer players	44	✅
All players human who ragequit during the game		✅

Only one player ragequits in the game		✅
All players human who play all commands		✅