Racko

This part of the homework walks you through completing all the functions you need to create that do not involve the computer's strategy.

We have supplied one possible computer strategy to you. If you want to work on the extra credit, you have to replace our code in computer\_play

You will be using LinkedLists (the inbuilt java class) to represent the deck and the discard pile. You will also need to use a method from the Collections library in order to shuffle cards. Copy and paste this at the top of your file to import the LinkedList and Collections libraries.

import java.util.LinkedList;

import java.util.Collections

Also, initialize your deck and discard as:

private static deck LinkedList<Card> deck;

private static discard LinkedList<Card> discard;

**Functions to be completed**

* load\_deck(LinkedList<Card> deck)

This function loads the deck you initialized with cards 1 through 60.

* shuffle()

This function shuffles the deck or the discard pile. To decide which one it needs to shuffle, it checks the length of the deck. It shuffles the deck to start the game. Also used to shuffle the discard pile if we ever run out of cards in the deck (length of deck being 0). To shuffle the deck you can call the shuffle method from the Collections library like this:

Collections.shuffle(deck);

* public static boolean check\_racko(Card[] rack)

Given a rack, either the user's or the computer's, determine if Racko has been achieved. Achieving Racko means that the cards are in ascending order.

* public static Card deal\_card()

This function gets the top card from the deck. It is used at the start of game play and again during each player's turn if they decide to not take the top card from the discard pile.

* public static Card[][] deal\_initial\_hands()

Start the game off by dealing two hands of 10 cards each. This function returns a 2D array; you will need to make it so that it can represent the user's hand and the computer's hand.

Make sure that you follow normal card game conventions of dealing. You have to deal one card to the user, one to the computer, one to the user, one to the computer and so on.

Remember that the rules of our version of Racko will be that you have to place your cards in the order of top most slot of the rack first, then the next slot and so on.

* print\_top\_to\_bottom

Given a rack print it out from top to bottom in a manner that looks more akin to the game (more stack like than list like). For example, given the rack: [2, 57, 53, 23, 45, 60, 12, 35, 7, 1], the rack will look like:

1

7

35

12

60

45

23

53

57

2

* public static boolean find\_and\_replace(Card newCard, int numberOfCard, Card[] rack)

This function finds the card that needs to be replaced in the hand, which is represented by the card's number/value, and then replaces it with the new card. The replaced card then gets put on top of the discard. Also, you want to check to make sure that the cardToBeReplaced is truly a card in the hand. If the user accidentally typed a card number incorrectly, leave the hand unchanged and have the user try again.

* public static void add\_card\_to\_discard(Card card)

Put the card in the discard pile.