CS220 - Arch from a Prog. Perspective
Project 1

Due: 04/11/2021, 11:59pm

1 Introduction

In this project, you will implement a program that evaluates the winner of a 2-player Texas Holdem' poker game. The card game Texas Holdem' poker is played using a standard 52 card deck where each player is dealt 2 cards and 5 shared ”community” cards. Each player's hand is the best (i.e., highest ranking) 5-card combination that can be formed out of the 7 cards: i.e., 2 player cards and 5 community cards. Hands are ranked, from lowest to highest, in the following way:

- **High Card:** Highest value card.
- **One Pair:** Two cards of the same value.
- **Two Pairs:** Two different pairs.
- **Three of a Kind:** Three cards of the same value.
- **Straight:** All cards are consecutive values.
- **Flush:** All cards of the same suit.
- **Full House:** Three of a kind and a pair.
- **Four of a Kind:** Four cards of the same value.
- **Straight Flush:** All cards are consecutive values of same suit.
- **Royal Flush:** Ten, Jack, Queen, King, Ace, in same suit.

The cards are valued in the order:
2, 3, 4, 5, 6, 7, 8, 9, 10, Jack, Queen, King, Ace.

If two players have the same ranked hands then the rank made up of the highest value wins; for example, a pair of eights beats a pair of fives (see example 1 below). But if two ranks tie, for example, both players have a pair of queens, then highest cards in each hand are compared (see example 4 below); if the highest cards tie then the next highest cards are compared, and so on.

A more elaborate description of poker hand rankings can be found here:
`http://www.pokerlistings.com/poker-hand-ranking`

Also, this YouTube video is informative:
`https://www.youtube.com/watch?v=PCPvJy6b4iA`

2 Goal

The goal of this project is to write a program that given the hands of 2 players, evaluates the winner. You will be provided with poker.c and poker.h files. You are to complete the specific tasks within the program, test it with the sample input and output.

Input: You will be provided with a plain text file that contains random hands dealt to two players. Each line of the file contains nine cards (each card separated by a single space): the first two are Player 1's cards, next two are Player 2's cards and the last five are community cards. Each players best hand constitutes the best 5 cards from the player's hand and the community cards combined. You can assume that all hands are valid (no invalid characters or repeated cards in the input).

Output: You are to generate Output.txt where each line contains the winner of the corresponding hand in the input.

3 Sample input and output

Table 1: Sample input and expected output

Hand	**Player 1**	**Player 2**	**Community**	**Expected Output**	**Reason**
1	5H 5C	8S 8D	3C 6D 7S TH QD	Player 2 wins	Pair of 8s beats pair of 5s
2	JS AC	KC JH	JC TD 2H 4S 6C	Player 1 wins	Pair of Jacks and High card Ace beats King
3	2C 4C	AH AS	TC 5C JD 4H AC	Player 1 wins	Flush beats 3 of a kind
4	AD 4C	KC KH	3D 3C 3H 3S KD	Player 1 wins	Four 3's and Ace beats four 3's and King
5	4C 4D	3C 3D	4S 3S 9D 9C AH	Player 1 wins	Full house with 3 fours beats full house with 3 threes.
6	9S 7H	9C 8S	AD 5H 9H AS JH	No single winner	Both players have 2 pairs Aces and 9s with Jack
7	AS AH	KC KD	AD 2H 3H 4S 5H	No single winner	Both players have a straight to 5

4 Provided Material

Other than this handout, you are provided with two files: `poker.c` and `poker.h` where each file has been marked with specific tasks you are supposed to implement. Additionally, you are provided with a reference implementation library (lib_poker_ref.a) that contains the reference implementation. In poker.c, you are to replace all calls to ref_*** functions with your own implementations. The reference implementation provides you a way to tackle each task independently. You are also provided with sample input and out output files with 100,000 hands. Your code may be tested on additional inputs.

5 Submitting the result

- You must maintain your code on github with regular updates.
- You are free to work on the project on remote.cs or on your laptops/desktop, however, ensure that the project compiles on remote.cs
- **You are required to demonstrate your code to your lab TA. Appointment slots will be made available close to the submission deadline.**
- Other than the demo to your lab TA, you are required to submit your final code on mycourses.