

ICS Homework 6

A Hand of Cards

For this assignment, you will build a deck of cards, draw several hands of cards from that deck, sort the hands in different ways, and print the results. You will use function pointers in order to customize the behavior of a single sort function that you write.

First, model a deck of cards. Create a typedef or structure to model a single card, and then create a deck of 52 such cards. A new deck should contain only one of each possible card, but in shuffled order. (Shuffling, like random number generation, is hard to do well. Consider [this blog posting](#) and [the Fisher-Yates shuffle algorithm](#).) You should get a different shuffled order each time you run your program or create a deck.

Next, model a hand of cards. You should be able to specify the maximum number of cards in each hand at the time you create it. For example, you might have a hand that can hold up to 5 cards (such as for draw poker) or up to 7 cards (such as for rummy). It would probably be helpful to also have a function that lets you add one or more cards from a deck to a given hand.

Write four card comparison functions. Each function should take two cards and return whether the first card would come before, after, or is equivalent to the second card. (This is usually done by returning a positive number, negative number, or zero.) The different comparison functions should use the following orders:

- by value only, Aces low (so A-K, disregarding suit)
- by value only, Aces high (so 2-A, disregarding suit)
- by suit, and then by value within a suit.
(The order of the suits and whether Ace is high in this ordering is up to you, but document your decision. Consider [various suit orders](#) for ideas.)
- a different order of your choosing. (Such as reversed order by value, a different suit order, etc.)

Now, write a function that sorts a given array of cards based on a given comparison function. You should only have ONE such sort function, but how it sorts is based on the passed pointer to a comparison function. (Note that, assuming you have a good design, it should also be possible to use this function to sort your entire deck, if you ever wanted to do so.) You must write your own sort function. (Do not use the stdlib's qsort.) This requirement is so that you get some practice using function pointers on the receiving end. The [sorting algorithm](#) you use is up to you: insertion, selection, bubble, quick, merge, heap, etc. (You should already know many of these from 211!)

Finally, write a separate function (which you will call from main) that does the following:

- Create a new shuffled deck of cards.
- Draw 2 hands of 5 cards each from that deck. Print them both sorted Aces low.
- Draw another 2 hands of 5 cards each. Print them both sorted Aces high.
- Draw another 2 hands of 7 cards each. Print them both sorted by suit.
- Draw a final 2 hands of a size of your choice. Print them in the ordering (not random) of your choice.

Use only your single card-sorting function for all of these sorts! Just pass a pointer to a different comparison function each time. Good luck!