This set of problems will lay the groundwork for our poker implementation.

GROUPS: There are three groups, which I will assign based on who I have seen working together (more or less). Please be sure that somebody in your group is comfortable with poker; if not, we can switch around as needed.

WHY ARE WE DOING THIS? The ability to write programs - even simple ones - strengthens you logically and, potentially, in terms of your eventual career. The ability to write and understand software sets you apart, and you have taken a step down that road. In the real world, a lot of software is developed by teams, and the coordination between and within such teams is key (outside of software development, too). So, here’s a flavor of that style of work for you. Besides, this will make you think about something bigger than the toy problems we have been tackling - this is a real solution to a real problem. Also, you get how Maple works now - what you know, paired with the excellent help menu, should serve you well in your future math courses. Finally, it gives you something to take home and play with (and extend) after our class ends this week....

GRADES: As for grading, 6 points today come from your group’s output. The other 4 come from what I perceive as your contribution. Not everybody will say something useful, but you need to keep your head in the game and actively work with your group. If I see you out in the hall on your phone for half the class, well....

ORGANIZATION: I have a master plan in mind and have implemented much of it on my own. I’ll dish out bits and pieces of the big picture for you to work on in your groups and piece them together between classes. It is essential that you follow exactly the plan that I lay out for you so that the pieces fit together well - especially be sure that the input and output is correct.

KEY POINTS:

1. We’ll use the integers 1 through 52 to represent the cards (no jokers).
2. The order of the suits is alphabetical: clubs, diamonds, hearts, spades.
3. The order of the cards in a suit is 2, 3, ..., 10, Jack, Queen, King, Ace.
4. The cards are ordered by suit and then by rank, i.e., the first card is the 2 of clubs, then the 3 of clubs, etc. Ace of spades is (fittingly) last.
5. Each hand will consist of five numbers (between 1 and 52) in a list, i.e., [15,2,30,16,9]. The two hands will be passed around together (when it makes sense to) as a list of two lists, i.e., [[15,2,30,16,9],[7,51,42,22,23]] with the user’s hand first and the CPU’s hand second.
6. There will be an array called draw_pile of size 52. An entry of 0 means that the card has not yet been drawn whereas an entry of 1 means that it has.
7. This is 5 card draw, with a single draw. In other words, a hand will be dealt to each player, there will be a chance for each player (in turn) to turn in k (up to 5) cards and get k new ones. After that, we’ll see who has the best hand and declare the winner.
8. Be sure to do some error checking in your procedures. If you get passed the number 53 as a card, you need to spit out an error.
9. Use descriptive names for your procedures and variables. You can call your procedure “myProc” and the variables x1, x2, x3, .... However, that won’t mean anything to anybody other than you (and not even you by next Tuesday!).

10. If your group gets done before another group, go ask if you can help somehow. Team 1 and Team 3 have brainstorming work to do, so you might be able to help with that, if nothing else....

WHAT TO TURN IN: Please send me your worksheet using RED TYPE (it’s ugly, but we need a standard since this is all going together in the end) - I want to get exactly three files (i.e., I don’t want one from everybody, just one per group). Documentation is really, really important, so please throw in some explanation of what you are doing.

ASSIGNMENTS:

Team 1

Your job today is the display of the cards in a human-friendly format. You need to have a procedure that takes in one hand (see the key points above) and spits out a list of the cards as “8 of hearts, etc.” Try to make it attractive and informative for the user.

I suggest (but don’t require) having two procedures (at least) – one for displaying a single card and one that displays the entire hand by calling the first procedure five times.

Be sure to test it on both good input and bad input. Part of working with other teams is catching their errors! If they pass you a 53, print out a big, ugly error.

The task so far is fairly easy compared to the other two teams. So, please think about making a graphical display of some sort. You’ll need lines, numbers, and symbols (or maybe something else).... This might prove to be too involved, but please at least brainstorm about it a bit and include some comments at the end of your worksheet about the ideas you had.

Team 2

Your job is to set up and populate the two hands (see the key points above). You’ll also need to cook up a procedure to take in a hand, the draw pile, and a list of cards to be replaced (indices, between 1 and 5) and, well, replace the cards.

First, you need to create draw_pile and fill it with 0s.

Then, you need to choose numbers for each of ten cards (without using cards that have been drawn!).

Next, for the last procedure, just grab a fresh card (randomly) for each card to be replaced.

Don’t forget to test your functions thoroughly.

Team 3

Given a pair of hands, your job is to write two procedures and to think.

First, you need a procedure that takes in a hand and arranges it in (numerical) order, i.e., 1,2,...,52. You
can use any sorting algorithm that you like - it doesn’t need to be slick.

Second, you need to write a procedure that asks the user how many cards they would like to replace and WHICH cards they want to discard (i.e., card 1 and card 4). Be sure the procedure returns a list of the indices of the cards being discarded.

Finally, think about how to work out the logic for the CPU’s hand. The user can specify which cards to replace, but what will we do for the CPU? We can’t get too sophisticated due to time and ability constraints, but we can certainly come up with something.... Please include a list of your brainstormed ideas at the bottom of your worksheet.