

CSE 341, Spring 2004, Assignment 1

Due: Wednesday 7 April, 9:00AM

Last updated: 26 March, 11AM

You will write five functions, all of which have something to do with playing cards. All necessary card knowledge is in this paragraph. Assume a playing card is represented by a *pair of integers*, the first integer is the *suit* and the second integer is the *value*. The suit should be a value between 1 and 4 (inclusive) and the value should be a value between 1 and 13 (inclusive). We say that a card “is a spade” if the suit is 4.¹ We say a card “is even” if its value is an even number.

1. Write a function `is_legal_card` that takes a pair of integers and evaluates to true if and only if the pair represents a “legal” card as explained above.
2. Write a function `is_all_spades` that takes a list of cards (legal or illegal) and evaluates to true if and only if every card in the list is a spade. (Note: If there are no cards in a list, then every card in the list is a spade.)
3. Write a function `get_all_even` that takes a list of cards (legal or illegal) and evaluates to a list of cards. A card should be in the result if and only if that card is in the input and that card is even. (Note: SML has a built-in infix operator `mod` that is like Java’s `%` operator. That is, `x mod 2` evaluates to 0 if and only if `x` is even.)
4. Write a function `biggest_spade_1` that takes a list of cards (legal or illegal) and evaluates to a list of integers. If no cards in the input are spades, then the result should be the empty list. Else the result should be a list with one integer, and that integer must be the value of the spade from the input that has the greatest value.

Hint: If there are no cards, the correct answer is the empty list. Otherwise, consider the value of the first card (call this `hd_num`) and the biggest spade in the tail of the list (call this `tl_ans`). If the first card is not a spade, then `tl_ans` is the right answer. If the first card is a spade and there are no spades in the rest of the list, then the list holding the first card’s value is the right answer. Else the right answer depends on which is greater: the first card’s value or the first (and only) element in `tl_ans`.

5. Write a function `biggest_spade_2` that is like `biggest_spade_1` except that it returns a value of type `int option`. (See next paragraph.)

It is bad style to use a list (which can hold any number of elements) when you will always want zero or one element. In Java, the constant `null` (not to be confused with ML’s function of the same name) is useful for the “zero” case.² In ML, we have “options”:

- If `t` is a type, then `t option` is a type.
- To make a value of type `t option`, write `NONE` (for “zero” elements) or `SOME e` (for “one” element) where `e` has type `t`.
- The function `isSome` evaluates to `true` if and only if its argument has the form `SOME e`.
- The function `valOf` evaluates to `e` when applied to `SOME e`. It raises an exception when applied to `NONE`.

6. **(Extra Credit)** Write a function `legal_deck` that takes a list of cards and evaluates to true if and only if all cards are legal (see problem 1) *and* no two cards in the list have the same suit and value. (Two cards can have the same suit or the same value, but not both.)

Note: Remember the course policy on extra credit!

¹Clubs 1, Diamonds 2, Hearts 3, but none of that matters for the homework. This is bad style we will fix next week.

²In fact, it is far *too* useful because there is no way in Java to indicate that a value should *not* be `null`.

Type Summary: Evaluating a correct homework solution should generate these bindings:

```
val is_legal_card = fn : int * int -> bool
val is_all_spades = fn : (int * int) list -> bool
val get_all_even = fn : (int * int) list -> (int * int) list
val biggest_spade_1 = fn : (int * int) list -> int list
val biggest_spade_2 = fn : (int * int) list -> int option
```

Of course, generating these bindings does not guarantee that your solutions are correct: *Test your functions.*

Assessment: Your solutions should be:

- Correct
- In good style, including indentation and line breaks
- Written using the features we have used in class. In particular, you must not use references or arrays. (Why would you?)

Turn-in Instructions

- Put all your solutions in one file, `lastname_hw1.sml`, where `lastname` is replaced with your last name.
- The first line of your `.sml` file should include an ML comment with your name and the phrase `homework 1`.
- Email your solution to `daverich@cs.washington.edu`.
- The subject of your email should be *exactly* `[cse341-hw1]`.
- Your `.sml` file should be an *attachment*.