More LISP

Input and output

- Read: reads from the input stream; interprets what it finds as a LISP object; returns it.
- Write: Writes its argument to the output stream.
- Format: Puts arguments into a control string and prints the result. Example:
  - (format t "I made $~$ this week. ~%You made $~$." 45.5 92)
- With-open-file: Politely opens files, and makes sure they're closed when you're done.
  - (with-open-file (stream "somefile2.txt" :direction :output) (format stream "some text"))
  - (with-open-file (stream "somefile2.txt") (read-line stream))

See http://www.tutorialspoint.com/lisp/lisp_input_output.htm for many variations on the theme, and http://www.gigamonkeys.com/book/files-and-file-io.html for an explanation of file handling.

More useful functions

- (defvar **global_variable** [value]). Note that the *s indicate that this really is meant to be a global variable.
- (setq *variable_name* value). Sets the value of a variable (quoted). Could be global, could be local.
- +, -, etc. Yes, you get math!
- (nth integer list). Returns the item at position *integer* in *list.*
- Other conditionals: Frankly, I recommend that you stick with *cond*, but there are other ways of handling conditionals: https://www.cs.cmu.edu/Groups/AI/html/cltl/clm/node84.html

Missionaries and cannibals problem

Three missionaries and three cannibals need to cross a river. There is one boat that can take two passengers. If the cannibals outnumber the missionaries on either shore, the missionaries get eaten (fail!).

1. How would you represent states in this puzzle? Hint: the puzzle says nothing about crocodiles or waterfall, so you can assume that setting out from one shore means reaching the other.
2. What are the goal/start states?
3. How would you define legal moves?

Assignment 2

- See “Resources” on Laulima