Instructions
Answer question #1 and any three others. (If you answer more, only the first four will count.) Point values are as indicated. Please try to make your answers neat and coherent. Remember, if we can’t read it, it’s wrong. Partial credit will be given, so try to put something down for each question (a blank answer always gets 0 points!).

1. (1 point)
What campus of the University of Wisconsin system is located in Madison?

2. (33 points)
Let \texttt{AllButLast} be a function that returns all of a string but its last character. For example, \texttt{AllButLast(abc) = ab}. \texttt{AllButLast(\lambda)} is undefined. Let \texttt{R} be any regular expression that does not generate \lambda. \texttt{AllButLast(R)} is the set of strings denoted by \texttt{R}, with \texttt{AllButLast} applied to each string. Thus \texttt{AllButLast(a* b) = a*}.

Show that \texttt{AllButLast(R)} is a regular set.

3. (33 points)
Assume we add a conditional compilation facility to CSX. The symbol \texttt{#if} will be followed by an intlit. If the intlit is non-zero all the characters up to a \texttt{#end} symbol will be scanned as usual. The \texttt{#if}, the intlit and the \texttt{#end} will be deleted (just like white space or comments). If the intlit following the \texttt{#if} is zero, all the characters up to the \texttt{#end} will be skipped, as will the \texttt{#if}, the intlit and the \texttt{#end}. Thus if we have
\begin{verbatim}
# if 1
   a = b+c;
# end
\end{verbatim}
the assignment will be scanned, but if we have
\begin{verbatim}
# if 0
   a = b+c;
# end
\end{verbatim}
the assignment will be skipped by the scanner.
Outline how you’d add conditional compilation to your CSX scanner. Should \texttt{#if} and \texttt{#end} pairs be allowed to nest? What must you do to support nested conditional compilation commands?
4. (33 points)
Assume we wish to write a program that examines a file that contains English language text (like a research paper or news report) and lists the 10 words that appear most often. Outline how you might use JLex to facilitate writing this program. What would you do to make your JLex definition ignore common articles and connectives (like “the” and “or” and “dude”)?

5. Write JLex regular expression definitions that match the following strings:
   (a) (11 points)
The four characters: "\n"

   (b) (11 points)
Any odd number of backslash characters (e.g., \ or \ \ or \ \ \ \ \, etc.).

   (c) (11 points)
A multi-line comment, delimited by { and }, that is allowed to contain no more than two new-line characters. That is, the comment may appear entirely on one line, or it may span two or three lines, but no more than three lines.

6. (33 points)
Let $S = \{ [ i ]^j | i \neq j \}$. $S$ is the set of all unbalanced brackets; that is, a number of left brackets followed by a different number of right brackets. Is $S$ a regular set? If it is, give a regular expression or finite automaton that defines it. If $S$ isn’t a regular set, explain carefully why.