1. (6 pts) What is the output (if any) of the following Ruby programs? Write FAIL if code does not execute.

   a. (2 pts)

       a = "maryland terps"
       # Output = Missed
       if (a =~ /y+land$/)
         puts "Found one #$1"
       else
         puts "Missed"
       end

   b. (2 pts)

       b = { "John" => 10, "Mary" => 20}
       # Output = nil
       puts b[10]
       puts b["Mary"]

   c. (2 pts)

       a = -1
       b = 1
       c = a + b
       if c
         puts "#{a} #{b}"
       else
         puts c
       end
       # Output = -1 1
2. (8 pts) Write a Ruby method \texttt{get\_tag} that given a string \texttt{str}, uses regular expressions and back references to find and return a valid car tag. A valid car has three lowercase characters, followed by dash (-), followed by three digits. For instance, \texttt{get\_tag("this is a tag akm-432 we use")} should return “akm-432”. The method will return “NoTag” if there are no tags in the input string.

\begin{verbatim}
def get_tag(str)
    if str =~ /([a-z]{3}-\d{3})/
        $1
    else
        "NoTag"
    end
end
\end{verbatim}

Alternative REs: \texttt{/(\([a-z]\)[0-9]\)/}, \texttt{/(\([a-z]\)[a-z]\)[0-9]\)/}, etc…

3. (6 pts) Write a Ruby method \texttt{square} that given an array of integers \texttt{int\_values}, uses the \texttt{Array.each} method and a code block to print each array value and its square. For instance, given the array [3, 2, 6, 1], your code should print out the following:

\begin{verbatim}
3 9
2 4
6 36
1 1
\end{verbatim}

\begin{table}
\centering
\begin{tabular}{|l|}
\hline
Some helpful functions (not all need to be used) \tabularnewline
\hline
\texttt{a.each \{ … \}} \tab \texttt{// apply code block to each element in array} \tabularnewline
\hline
\texttt{puts b} \tab \texttt{// print b followed by a newline} \tabularnewline
\hline
\texttt{c.to\_s} \tab \texttt{// returns string for c} \tabularnewline
\hline
\end{tabular}
\end{table}

\begin{verbatim}
def square(int\_values)
    int\_values.each { |x|
        y = x * x
        puts "#{x} #{y}"
    }
end
\end{verbatim}