1 Generating Vanity Plates

Write an Icon program to generate vanity plates, car license plates with cute messages. The idea is to try to shorten long words taken from a standard dictionary by replacing sequences of letters with shorter character sequences. The resulting license plate must consist of at most six letters and digits. Here are some examples:

<table>
<thead>
<tr>
<th>Original Word</th>
<th>License Plate</th>
<th>Rules Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>fortunate</td>
<td>4TUN8</td>
<td>for → 4, ate → 8</td>
</tr>
<tr>
<td>stench</td>
<td>S10CH</td>
<td>ten → 10</td>
</tr>
<tr>
<td>ozone</td>
<td>OZ1</td>
<td>one → 1</td>
</tr>
<tr>
<td>mezzanine</td>
<td>MEZZA9</td>
<td>nine → 9</td>
</tr>
<tr>
<td>foursquare</td>
<td>4SQARE</td>
<td>four → 4, qu → q</td>
</tr>
<tr>
<td>forklift</td>
<td>4KLIFT</td>
<td>for → 4</td>
</tr>
<tr>
<td>european</td>
<td>EUROPN</td>
<td>pea → p</td>
</tr>
<tr>
<td>detour</td>
<td>DE2UR</td>
<td>to → 2</td>
</tr>
</tbody>
</table>

The input of your program should be a sequence of words, one per line, read from standard input.

The output of your program should be a sequence of pairs of the form \textit{original word} → \textit{vanity plate}, like this:

\begin{verbatim}
fortunate --> 4tun8
stench    --> s10ch
ozone     --> oz1
\end{verbatim}

Your program should make judicious use of Icon’s

- \textit{generators} (i.e. procedures that generate multiple results using the \texttt{suspend} statement),
- built-in data structures (i.e. lists, records, sets, and tables), and
- \textit{goal-directed evaluation} (i.e. bounded backtracking).

There are many collections of vanity plates on the internet that can give you ideas for constructing your rule base. See, for example, \url{http://www-chaos.umd.edu/misc}. 
2 Code Template

Here is a template for the structure of your Icon program:

```
# produce all possible vanity-plates from the word
# 'str', given the possible substitutions in the table
# 'substitutions'.
procedure generate(str, substitutions)
    ....
    suspend result
    ....
end

# read all words from standard input into a list, and
# return this list
procedure readWords()
    ....
    return words
end

# build a table of possible substitutions, such as
# "one" --> "1", and return this table.
procedure buildTable()
    ...
    return tab
end

procedure main(args)
    words := readWords()
    substitutions := buildTable()
    every input := !words do
        every st := generate(input, substitutions) do
            write(input, " --> ", st)
    end
end
```

3 Submission and Assessment

The deadline for this assignment is Noon, Wed Dec 1. You should submit the assignment electronically using the Unix command `turnin cs372.9 <files>`.

This assignment is worth 4% of your final grade.

Name your program `vanity.icn`.

Don’t show your code to anyone, don’t read anyone else’s code, don’t discuss the details of your code with anyone. If you need help with the assignment see the instructor.