

6 Control Flow

6.9 Exercises

- 6.34 (David Hanson [Han93].) Write a program in Icon that will print the k most common words in its input, one per line, with each preceded by a count of the number of times it appears. If parameter k is not specified on the command line, use 10 by default. You will want to consult the Icon manual (available on-line [GG96]). In addition to `suspend`, `upto`, and `write`, discussed here, you may find it helpful to learn about `integer`, `many`, `pull`, `read`, `sort`, `table`, and `tab`. When fed the Gettysburg Address, your program should print:

```
13  that
9   the
8   we
8   to
8   here
7   a
6   and
5   of
5   nation
5   have
```

- 6.35 Write a `findRE` generator in Icon that mimics the behavior of `find`, but takes as its first parameter a regular expression. Use a string to represent your regular expression, with syntax as in Section 2.1.1. Use empty parentheses to represent ϵ . Give highest precedence to Kleene closure, then concatenation, then alternation. You may assume that we never search for vertical bar, asterisk, or parenthesis characters.

6.36 Explain why the following guarded commands in SR are *not* equivalent:

<code>if a < b -> c := a</code>	<code>if a < b -> c := a</code>
<code>[] b < c -> c := b</code>	<code>[] b < c -> c := b</code>
<code>[] else -> c := d</code>	<code>[] true -> c := d</code>
<code>fi</code>	<code>fi</code>

6.37 Write, in SR or pseudocode, a function that returns

- (a) an arbitrary nonzero element of a given array
- (b) an arbitrary permutation of a given array

In each case, write your code in such a way that if the implementation of nondeterminism were truly random, all correct answers would be equally likely.