

```
1: #! /Users/jj/bin/perl5.10.0
2:
3: use 5.010;
4: use strict;
5: use warnings;
6:
7: my @tests = (
8:     make_rule('Password is in word list',
9:         sub {/^$_$/i ~~ wordlist()} ),
10:
11:     make_rule('Password minus last letter is in word list',
12:         sub {chop; /^$_$/i ~~ wordlist();} ),
13:
14:     make_rule('Password minus first letter is in word list',
15:         sub {$_ = substr $_,1; /^$_$/i ~~ wordlist();} ),
16:
17:     make_rule('Substitutes "0" (zero) for letter "o"',
18:         sub {s/o/o/g and /^$_$/i ~~ wordlist();} ),
19:
20:     make_rule('Substitutes "1" (one) for letter "l"',
21:         sub {s/l/l/g and /^$_$/i ~~ wordlist();} ),
22:
23:     make_rule('Length is not between 10 and 20 characters',
24:         sub{ length($_) < 10 or length($_) > 20 } ),
25:
26:     make_rule('Password does not include a digit',
27:         sub {$_ !~ /\d/i} ),
28:
29:     make_rule('Password does not include an uppercase letter',
30:         sub {$_ !~ /[A-Z]/i} ),
31:
32:     make_rule('Password does not include a lowercase letter',
33:         sub {$_ !~ /[a-z]/i} ),
34:
35:     make_rule('Password does not include a symbol',
36:         sub {$_ !~ /^[^0-9A-Z]/i} ),
37:
38:     make_rule('Password has 4 lowercase letters in succession',
39:         sub {$_ =~ /[a-z]{4}/i} ),
40:
41:     make_rule('Password has 4 uppercase letters in succession',
42:         sub {$_ =~ /[A-Z]{4}/i} ),
43:
44:     make_rule('Password has duplicate characters',
45:         sub {$_ =~ /(.)*\1/i} ),
46: );
47:
48: say "Enter a password to test:";
49: my $password = <STDIN>;
50: chomp $password;
51:
52: my $score = @tests;
53: foreach my $test (@tests) {
54:     if (my $message = $test->($password)) {
55:         $score--;
56:         say $message;
57:     }
58: }
59:
60: say '';
61:
62: given ($score) {
```

```
63:  when ($_ >= 11) {
64:      say "A score of $score indicates a strong password";
65:  }
66:  when ([7..10]) {
67:      say "A score of $score indicates a moderately strong password";
68:  }
69:  when ($_ <= 6) {
70:      say "A score of $score indicates a weak password";
71:  }
72: }
73:
74:
75: #-----
76:
77: sub make_rule {
78:     # Creates a function to test $_ against the specified rule,
79:     # returning $message if the rule matches
80:     my ($message,$rule) = @_;
81:     return sub {
82:         local $_ = shift;
83:         return ($rule->) ? $message : undef;
84:     }
85: }
86:
87: #-----
88:
89: sub wordlist {
90:     state @wordlist;
91:     unless (@wordlist) {
92:         open WORDLIST,'<','wordlist.txt' or die "Cannot open wordlist.txt: $!";
93:         @wordlist = map {s/\s//g; $_} (<WORDLIST>);
94:         close WORDLIST;
95:     }
96:     return \@wordlist;
97: }
98:
99: #-----
```