

(move) ← Move about to be mutated
BLUE ← Resulting mutation
GREEN ← Remaining moves as a result of mutation

```
[1]> (load "pegs.1")
;; Loading file pegs.1 ...
;; Loaded file pegs.1
T
[2]> ( setf p (play-full) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((3 3 *) L)
((3 0 *) R) ((1 1 *) DR) ((4 0 *) R) ((3 3 *) L) ((4 2 *) UL))
[3]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((4 0 *) UR)
((3 3 *) L) ((2 0 *) DR) ((1 1 *) DR) ((4 2 *) L) ((4 4 *) UL))
[4]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((4 0 *) UR)
((3 3 *) L) ((2 0 *) DR) ((1 1 *) DR) ((4 4 *) UL) ((4 1 *) R))
[5]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((4 0 *) UR)
((3 3 *) L) ((1 1 *) DR) ((4 1 *) UR) ((2 0 *) R) ((3 3 *) UL))
[7]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((4 0 *) UR)
((3 3 *) L) ((1 1 *) DR) ((4 1 *) UR) ((4 4 *) UL))
[10]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((1 0 *) DR) ((4 3 *) L) ((3 3 *) L)
((1 1 *) DR) ((4 0 *) UR) ((4 4 *) UL) ((2 0 *) DR))
[11]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((4 3 *) L) ((1 0 *) DR) ((4 0 *) R)
((3 3 *) L) ((3 0 *) R) ((1 1 *) DR) ((3 3 *) L))
[12]> ( setf p (mutate p) )
(((2 0 *) UR) ((3 2 *) UL) ((4 1 *) UR) ((4 3 *) L) ((4 0 *) UR) ((1 1 *) DL)
((3 3 *) UL) ((1 0 *) DL) ((4 1 *) UR) ((0 0 *) DR))
[13]> (bye)
Bye.
```