

Bemerkenswert ist hier, dass LOGO als Funktionswert eine ganze Liste abgeben kann.

Die Prozedur loesung [25] gibt einen Kommentar aus und ruft die Prozedur prloesung [26] auf, die eine Lösung mit einer Karte pro Zeile ausgibt. Typisch für LOGO ist die rekursive Verarbeitung der Liste.

(Bemerkung: Da LOGO eine interpretierte Sprache ist, läuft das Programm recht lange. Sie sollten mit einigen Stunden rechnen!)

Anmerkung der Red.:

Das publizierte Kartenspiel besitzt (abgesehen von der Rotationsymmetrie) folgende 12 Lösungen:

(zeilenweise (durch Komma abgetrennt): Kartennummer, anschließend Lage der Nummer auf der Karte: oben (o), rechts (r), unten (u), links (l))

1. 112r8r,314r9u,5r6u7r
2. 2r3o5u,4r9u1o,6u7r8r
3. 2r3o5u,4r9u1o,7r8r6u
4. 2r3o5u,4r9u1o,8r6u7r
5. 2r8r6u,4r9u1o,5u3o7r
6. 2r8r6u,4r9u1o,7r3o5u
7. 2u7u4u,3r9l8u,5llr6l
8. 2u7u4u,8u9l3r,6llr5l
9. 2u8u4u,6l9l3r,7ulr5l
10. 4u2u7u,3r9l8u,5llr6l
11. 4u2u7u,8u9l3r,6llr5l
12. 4u2u8u,6l9l3r,7ulr5l

LOGO Programcode

```
MAKE "schildkroeten' >
[[1 0 [f2 k1 t1 k2]] [2 0 [f1 k1 t2 s2]] [3 0 [t2 k1 s1 f2]] >
[4 0 [f1 t1 s2 k2]] [5 0 [f1 k2 t2 t1]] [6 0 [f1 f2 t2 k1]] >
[7 0 [f1 s1 k2 t2]] [8 0 [f1 s1 f2 t2]] [9 0 [t1 f2 s2 k1]]]]

TO suche? :s
PR [Ich beginne zu suchen]
suchen :s [] []
PR []
PR [Das Programm ist zu Ende]
END
```