

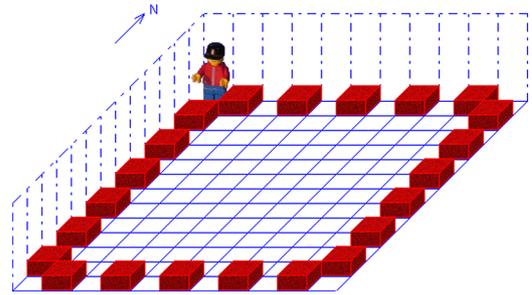
Karol soll einen Rahmen um seine Welt legen, aber nur jeden zweiten Stein hinlegen.

### 1. Versuch

```
Anweisung Seite ()
  wiederhole solange nicht IstWand
    Hinlegen()
    Schritt()
    Schritt()
  *wiederhole
  LinksDrehen()
*Anweisung
```

### Programm

```
wiederhole 4 mal
  Seite ()
  *wiederhole
*Programm
```



### Problem:

*Wenn die Länge oder Breite der Welt geradzahlig ist, bricht das Programm ab!*

*Der 2. Schritt darf nur dann ausgeführt werden, wenn Karol noch nicht vor der Wand steht!*

### 2. Versuch

```
Anweisung Seite ()
  wiederhole solange nicht IstWand
    Hinlegen()
    Schritt()
    wenn nicht IstWand
      dann Schritt ()
    *wenn
  *wiederhole
  LinksDrehen()
*Anweisung
```

### Programm

```
wiederhole 4 mal
  Seite ()
  *wiederhole
*Programm
```

Weiteres Beispiel

Karol soll eine Reihe von Steinen „invertieren“, d.h. dort, wo ein Stein liegt, soll dieser aufgehoben werden und dort, wo kein Stein liegt, soll einer hingelegt werden.

```
Anweisung Invertiere ()
  wenn istZiegel
    dann aufheben ()
  sonst hinlegen ()
*wenn
```

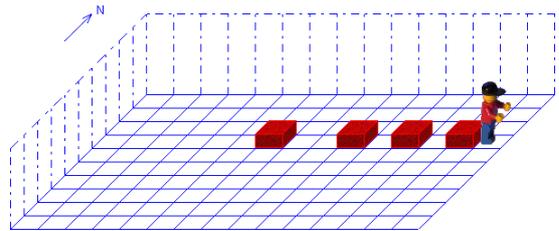
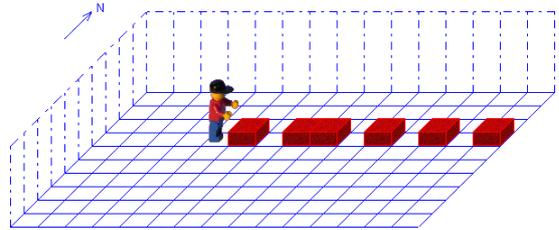
\*Anweisung

Programm

```
Wiederhole solange nicht IstWand
  invertiere ()
  Schritt ()
```

```
*wiederhole
```

\*Programm



Die vollständige Form der bedingten Anweisung

```
wenn Bedingung
  dann Sequenz_1
  sonst Sequenz_2
*wenn
```

*d.h.: wenn die Bedingung richtig (wahr) ist, dann wird Sequenz\_1 ausgeführt, wenn nicht, dann Sequenz\_2.*

*In manchen Fällen (siehe 1. Beispiel) ist nur in einem Fall eine Sequenz auszuführen. Dann kann der Teil sonst Sequenz\_2 entfallen.*