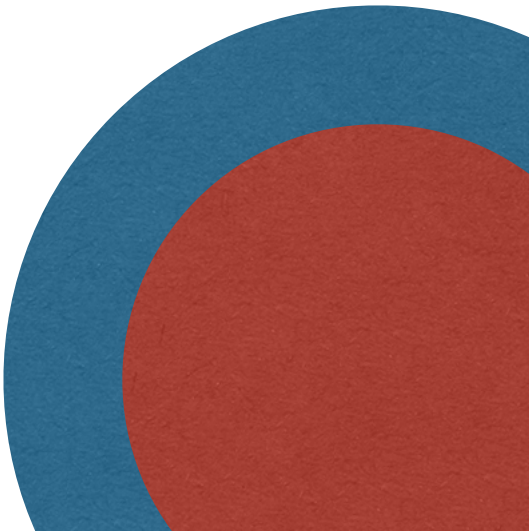
42. Bundeswettbewerb Informatik

Team Deutscher Qualitätscode
Max Eckstein, Nina Hochecker
Team-ID: 00538



Aufgabe 4: Nandu

Inhaltsverzeichnis

Lösungsidee

Wir haben die Aufgabe in zwei Teile unterteilt.
Im ersten Teil geht es um die Ermittlung der Status der PrüfLEDs in Abhängigkeit von den aktivierten Lichtquellen. Dafür müssen alle Bausteine nacheinander einzeln betrachtet werden, angefangen mit denen in der obersten Zeile. Der betrachtete Baustein muss Licht in Abhängigkeit von den Zuständen der Blöcke/Lichtquellen über sich weitergeben. An den PrüfLEDs kann schließlich ausgelesen werden, wie die PrüfLED Zustände für die gegebene Lichtquellenkonfiguration ist.

Im zweiten Teil muss für jede mögliche Konfiguration der Lichtquellen der obige Prozess durchgeführt und ausgewertet werden.

Wir haben uns außerdem dazu entschieden, dass Licht nur von Block zu Block übertragen werden kann, und nicht gradlinig im Raum weiterläuft, wenn grade kein Baustein vorhanden ist. Dies reduziert die Komplexität der Aufgabe drastisch. Zudem entspricht dies dem erwarteten Ergebnis im echten Leben.

Umsetzung

Für die verschiedenen Blöcke (weiße-, rote- und blaue Bausteine, Lichtquellen und PrüfLEDs) haben wir verschiedene Structs erstellt. Alle diese Structs haben jeweils einen Höhen- und einen Breitenindex, welcher ihre Position auf dem Spielfeld angibt. Ganz oben links (in den Dateien) ist der Ursprung (Höhenindex = 0, Breitenindex = 0).
Bei den Bausteinen repräsentieren diese Koordinaten die Koordinaten des linken Teils des Bausteins. Alle Bausteine haben zusätzlich die Funktionen linkesOutput und rechtesOutput, welche mithilfe von den zwei Parametern linkesInput und rechtesInput Bausteinspezifisch errechnen können, ob und wo der Baustein Licht abgibt. PrüfLEDs und Lichtquellen haben noch einen Zustand aktiv, welcher angibt, ob sie gerade leuchten. Initial ist dieser immer false.
Nachdem die externe Datei eingelesen wurde, kann aus der ersten Zeile die Breite und die Höhe des Spielfeldes ausgelesen werden. Das Array bausteine wird nun mit den in der Datei vorgegebenen Bausteinen befüllt. Außerdem werden auch die PrüfLEDs und die Lichtquellen aus der Datei ausgelesen.
Dazu wird jedes Zeichen in den Zeilen der Datei mit einem zweidimensionalen For-Loop betrachtet, und je nach Zeichen wird ein/e passender Baustein/Lichtquellen/PrüfLED in das korrespondierende Array hinzugefügt. Der äußere For-Loop iteriert durch die Zeilen, der Innere jeweils durch die Zeichen in der betrachteten Zeile. Der Index der Zeile wird

als Höhenindex für die Bausteine/Lichtquellen/PrüfLEDs benutzt. Bei dem Breitenindex verhält es sich etwas komplexer. Dieser wird nicht bei jedem Durchlauf der inneren For-Schleife erhöht, sondern, nur wenn gerade ein „X" gefunden wurde. Außerdem wird er um zwei erhöht, wenn ein Baustein in der Zeile entdeckt wird. Die Variable skippeDasNaechste wird benutzt, um anzuzeigen, ob der nächste passende Buchstabe als Baustein interpretiert werden soll. Initial ist sie false. Dies ist deshalb nötig, weil sonst bei dem Input „B B" zwei Bausteine hinzugefügt werden würden. Wenn also ein Baustein entdeckt wird, so wird skippeDasNaechste auf true gesetzt. Wenn das nächste Mal nun ein „neuer" Baustein (also eigentlich derselbe von davor) erkannt wird, wird skippeDasNaechste zurück auf false gesetzt und kein neuer Baustein zum Array hinzugefügt. Bei roten Bausteinen ist noch relevant, ob zuerst das kleine oder das große R erkannt wird, da festgelegt werden muss, ob sich der Sensor links oder rechts befindet. Nachdem alle Bausteine, Lichtquellen und PrüfLEDs hinzugefügt wurden, wird sichergestellt, dass die Anzahl der Lichtquellen sowie die Anzahl der PrüfLEDs ungleich null ist.

Anschließend wird die Funktion befuelleLichtkarte definiert, welche die Simulation von der Lichtverteilung durch die Bausteine hindurch durgeführt sowie den Status der PrüfLEDs ermittelt, basierend auf der aktuellen Konfiguration der Lichtquellen. Die Lichtverteilung auf dem Feld wird durch die Variable lichtkarte repräsentiert, welche ein zwei-dimensionales boolean Array ist. Die Funktion setzt zuerst die Lichtkarte zurück, indem alle Werte in der lichtkarte auf false gesetzt werden. Die Dimensionen der lichtkarte werden dabei durch die Werte von der Breite und Höhe bestimmt. Nun werden die Positionen der aktiven Lichtquellen auf der lichtkarte auf true gesetzt, da an dieser Stelle ja Licht wäre. Damit alle Bausteine das Licht weiterleiten, wird in einer Schleife durch alle Bausteine iteriert:

Nachdem geprüft wurde, dass sich der Baustein nicht am ganz oberen Rand befindet, wird der Wert des rechten/linken Inputs aus der darüber liegenden Zeile der Lichtkarte ausgelesen. Mithilfe der Funktion linkes- bzw. rechtesOutput wird der linke/rechte Output des Bausteins ermittelt und auf der Lichtkarte festgelegt.

Abschließend wird der Status der PrüfLEDs ermittelt, indem die lichtkarte in der Zeile über ihnen ausgelesen wird. Dieses „Ergebnis" der PrüfLEDs wird zusammen mit der aktuellen Lichtquellen Konfiguration zum Output hinzugefügt.

Die letzte Herausforderung ist die Generierung aller unterschiedlichen Konfigurationen der Lichtquellen. Konkret müssen alle unterschiedlichen Fälle aus einer korrespondierenden Warheitstafel generieret werden. Die Funktion testeFall übernimmt diese Aufgabe. Die Funktion nimmt zwei Parameter entgegen: einen Index, welcher initial

0 ist und ein Array von Lichtquellen, welches initial das aus der Datei ausgelesene ist. Die Funktion prüft, ob der ihr übergebene Index der letzte Index des übergegeben lichtquellen Array ist. Wenn dies der Fall ist, wird die Funktion befuelleLichtkarte mit der übergebenen Lichtquellenonfiguration aufgerufen. Außerdem wird befuelleLichtkarte nochmal mit einer abgeänderten Lichtquellenkonfiguration aufgerufen, bei der die letzte Lichtquelle jedoch angeschaltet ist. Wenn dies aber nicht der Fall ist, dann ruft die Funktion sich rekursiv selber auf, mit der übergebenen Lichtquellenkonfiguration und dem eigenen Index +1 als Parameter. Außerdem ruft die Funktion sich selbst noch mit einer abgewandelten Lichtquellenkonfiguration auf. Bei dieser wurde an der Stelle des übergebenen Index die Lichtquelle angeschaltet.

Abschließend werden die Ergebnisse, die in der Variable **output** gespeichert sind, in die Konsole ausgegeben.

Ausführung des Algorithmus:
In der Kommandozeile wird (auf MacOS) "./main" eingegeben, gefolgt von einem Leerzeichen und der Nummer der Datei, die ausgelesen werden soll. Bei „./main 4" wird beispielsweise die Datei „../Daten/nandu4.txt" ausgelesen.

Beispiele

Datei nadu1.txt:

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv	L1 ist aktiv L2 ist aktiv
Q1 ist inaktiv Q2 ist aktiv	L1 ist aktiv L2 ist aktiv
Q1 ist aktiv Q2 ist inaktiv	L1 ist aktiv L2 ist aktiv
Q1 ist aktiv Q2 ist aktiv	L1 ist inaktiv L2 ist inaktiv

Datei nadu2.txt:

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv	L1 ist inaktiv L2 ist aktiv
Q1 ist inaktiv Q2 ist aktiv	L1 ist inaktiv L2 ist aktiv
Q1 ist aktiv Q2 ist inaktiv	L1 ist inaktiv L2 ist aktiv
Q1 ist aktiv Q2 ist aktiv	L1 ist aktiv L2 ist inaktiv

Datei nadu3.txt:

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist inaktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv	L1 ist inaktiv L2 ist aktiv L3 ist inaktiv L4 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv	L1 ist inaktiv L2 ist aktiv L3 ist inaktiv L4 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv	L1 ist inaktiv L2 ist aktiv L3 ist aktiv L4 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv	L1 ist inaktiv L2 ist aktiv L3 ist aktiv L4 ist inaktiv

Datei: nadu4.txt:

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv	L1 ist aktiv L2 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv	L1 ist aktiv L2 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv	L1 ist aktiv L2 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv	L1 ist aktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist aktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv	L1 ist inaktiv L2 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv	L1 ist inaktiv L2 ist inaktiv

Datei nadu5.txt:

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist inaktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist inaktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist inaktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist inaktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist inaktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist inaktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist aktiv L4 ist inaktiv L5 ist inaktiv

Konfiguration	Output
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist inaktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv
Q1 ist aktiv Q2 ist aktiv Q3 ist aktiv Q4 ist aktiv Q5 ist aktiv Q6 ist aktiv	L1 ist aktiv L2 ist inaktiv L3 ist inaktiv L4 ist aktiv L5 ist aktiv

Quellcode

```
import Foundation

protocol Baustein {
    var hoehenIndex: Int { get }
    var breitenIndex: Int { get }
    func linkesOutput (l linkesInput: Bool, r rechtesInput: Bool) -> Bool
    func rechtesOutput (l linkesInput: Bool, r rechtesInput: Bool) -> Bool
}

struct WeisserBaustein: Baustein {
    var hoehenIndex: Int
    var breitenIndex: Int
    func linkesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return !(linkesInput && rechtesInput)
    }
    func rechtesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return !(linkesInput && rechtesInput)
    }
}

struct RoterBaustein: Baustein {
    var hoehenIndex: Int
    var breitenIndex: Int
    var sensorIsLinks: Bool
    func linkesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return sensorIsLinks ? !linkesInput : !rechtesInput
    }
    func rechtesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return sensorIsLinks ? !linkesInput : !rechtesInput
    }
}

struct BlauerBaustein: Baustein {
    var hoehenIndex: Int
    var breitenIndex: Int
    func linkesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return linkesInput
    }
    func rechtesOutput(l linkesInput: Bool, r rechtesInput: Bool) -> Bool {
        return rechtesInput
    }
}

struct Lichtquelle {
    let hoehenIndex: Int
    let breitenIndex: Int
    var aktiv: Bool = false
    func angeschaltet() -> Lichtquelle {
        return Lichtquelle(hoehenIndex: self.hoehenIndex, breitenIndex: self.breitenIndex, aktiv:
true)
    }
}

struct PruefLED {
    let hoehenIndex: Int
    let breitenIndex: Int
    var aktiv: Bool = false
}

// Einlesen der externen Datei und befüllen der Baustein-Liste, Eintragen der Positionen der Prüf-
LEDs und der Lichtquellen
let dateiNummer: String = CommandLine.arguments.last ?? "-"
let dateiNummerInt: Int = Int(dateiNummer) ?? 1
let pfad: URL = URL(fileURLWithPath: "../Daten/A4_Nandu/nandu\(dateiNummerInt).txt")
guard let text: String = try? String(contentsOf: pfad) else {
    print("Datei konnte nicht gefunden / ausgelesen werden")
    exit(EXIT_FAILURE)
}
var zeilen = text.split(whereSeparator: \.isNewline)
let ersteZeile = zeilen.removeFirst()

guard let breite = Int(ersteZeile.split(separator: " ")[0]), let hoehe =
Int(ersteZeile.split(separator: " ")[1]) else {
    print("Breite und Hoehe konnten nicht ausgelesen werden")
    exit(EXIT_FAILURE)
}
```

```
// Befüllen der Variablen

var bausteine: [any Baustein] = []
var S_pruefLEDs: [PruefLED] = []
var S_lichtquellen: [Lichtquelle] = []

var hoehenIndex = 0
var breitenIndex = 0

var skippeDasNaechste = false

for zeile in zeilen {
    for zeichen in zeile {
        switch zeichen {
            case "X":
                breitenIndex += 1
            case "W":
                if !skippeDasNaechste {
                    let neuerBaustein = WeisserBaustein(hoehenIndex: hoehenIndex, breitenIndex:
breitenIndex)
                    bausteine.append(neuerBaustein)
                    breitenIndex += 2
                    skippeDasNaechste = true
                } else {
                    skippeDasNaechste = false
                }
            case "B":
                if !skippeDasNaechste {
                    let neuerBaustein = BlauerBaustein(hoehenIndex: hoehenIndex, breitenIndex:
breitenIndex)
                    bausteine.append(neuerBaustein)
                    breitenIndex += 2
                    skippeDasNaechste = true
                } else {
                    skippeDasNaechste = false
                }
            case "R":
                if !skippeDasNaechste {
                    let neuerBaustein = RoterBaustein(hoehenIndex: hoehenIndex, breitenIndex:
breitenIndex, sensorIsLinks: true)
                    bausteine.append(neuerBaustein)
                    breitenIndex += 2
                    skippeDasNaechste = true
                } else {
                    skippeDasNaechste = false
                }
            case "r":
                if !skippeDasNaechste {
                    let neuerBaustein = RoterBaustein(hoehenIndex: hoehenIndex, breitenIndex:
breitenIndex, sensorIsLinks: false)
                    bausteine.append(neuerBaustein)
                    breitenIndex += 2
                    skippeDasNaechste = true
                } else {
                    skippeDasNaechste = false
                }
            case "Q":
                let neueLichtquelle = Lichtquelle(hoehenIndex: hoehenIndex, breitenIndex:
breitenIndex)
                S_lichtquellen.append(neueLichtquelle)
                breitenIndex += 1
            case "L":
                let neuePruefLED = PruefLED(hoehenIndex: hoehenIndex, breitenIndex: breitenIndex)
                S_pruefLEDs.append(neuePruefLED)
                breitenIndex += 1
            default:
                continue
        }
    }
    hoehenIndex += 1
    breitenIndex = 0
}

guard !S_lichtquellen.isEmpty && !S_pruefLEDs.isEmpty else {
    print("Es gibt keine Lichtquellen und/oder keine Prüf-LED's.")
    exit(EXIT_FAILURE)
}

var output: [([Lichtquelle], [PruefLED])] = []

var lichtkarte: [[Bool]] = []
```

```
func befuelleLichtkarte(lichtquellen: [Lichtquelle]) {

    // Lichtkarte zurücksetzen
    lichtkarte = [[Bool]](repeating: [Bool](repeating: false, count: breite), count: hoehe)

    // Lichkarte erhellen, an den Stellen an denen aktive Lichtquellen sind
    for lichtquelle in lichtquellen {
        lichtkarte[lichtquelle.hoehenIndex][lichtquelle.breitenIndex] = lichtquelle.aktiv
    }

    // Alle Bausteine das Licht weiterleiten lassen
    for baustein in bausteine {

        // Sicherstellen, dass der Baustein nicht ganz am oberen Rand ist
        guard baustein.hoehenIndex != 0 else { continue }

        let linkesInput = lichtkarte[baustein.hoehenIndex - 1][baustein.breitenIndex]
        let rechtesInput = lichtkarte[baustein.hoehenIndex - 1][baustein.breitenIndex + 1]

        lichtkarte[baustein.hoehenIndex][baustein.breitenIndex] = baustein.linkesOutput(l:
linkesInput, r: rechtesInput)
        lichtkarte[baustein.hoehenIndex][baustein.breitenIndex + 1] = baustein.rechtesOutput(l:
linkesInput, r: rechtesInput)
    }

    let pruefLEDs: [PruefLED] = S_pruefLEDs.map({
        var neueLED = $0
        neueLED.aktiv = lichtkarte[$0.hoehenIndex - 1][$0.breitenIndex]
        return neueLED
    })

    output.append((lichtquellen, pruefLEDs))
}

func testeFall(_ lichtquellen: [Lichtquelle], eigenerIndex: Int){
    let esGibtNochEinenWeiteren = eigenerIndex + 1 < S_lichtquellen.count
    if esGibtNochEinenWeiteren {
        testeFall(lichtquellen, eigenerIndex: eigenerIndex + 1)
        var neueLichtquellenKonfig = lichtquellen
        neueLichtquellenKonfig[eigenerIndex] = lichtquellen[eigenerIndex].angeschaltet()
        testeFall(neueLichtquellenKonfig, eigenerIndex: eigenerIndex + 1)
    } else {
        befuelleLichtkarte(lichtquellen: lichtquellen)
        var neueLichtquellenKonfig = lichtquellen
        neueLichtquellenKonfig[eigenerIndex] = lichtquellen[eigenerIndex].angeschaltet()
        befuelleLichtkarte(lichtquellen: neueLichtquellenKonfig)
    }
}

testeFall(S_lichtquellen, eigenerIndex: 0)

for i in output {
    print("Für Konfiguration:")
    for i in i.0.enumerated() {
        print("Q\(i.offset + 1) ist \(i.element.aktiv ? "aktiv" : "inaktiv")")
    }
    print("Kommt bei den Sensoren am Ende an:")
    for i in i.1.enumerated() {
        print("L\(i.offset + 1) ist \(i.element.aktiv ? "aktiv" : "inaktiv")")
    }
    print("---")
}
```