



Es reicht vermutlich, nur eine von beiden Aufgaben zu lösen. Langfristig ist es aber doch ziemlich cool, beide zu lösen. Also eigentlich sollte man beide lösen. Nur vielleicht noch nicht heute. Vielleicht aber auch schon.

Aufgabe 1. Schreibe ein Programm, das ein Labyrinth aus einer Datei einliest:

```
XXXXXXXXXXXXXXXXXX
X X XXXXXXXXXXXX*X
X$X XX      XXX X
X X XX XXX XXX X
X  XX XXX XXX X
XXX X  XX XXX X
XXX  X      X
XXXXXXXXXXXXXXXXXX
```

Bemerkung: Wir spezifizieren das Labyrinth hier nicht viel näher, entscheide dich selbst vorher für ein Format. Soll die Größe des Labyrinths variabel sein oder fest? Soll die Größe in der ersten Zeile der Datei stehen oder nicht? Soll das Labyrinth quadratisch sein oder nicht? Soll es außen herum immer mit Xen begrenzt sein oder hast du vielleicht eine andere Lösung?

Das Programm soll einen Weg vom Startpunkt (dem Stern) zum Schatz (dem Dollarzeichen) finden. Die Xe sind Wände und Leerzeichen sind Pfade. Markiere einen Weg mit Punkten und gebe das Labyrinth mit Weg in der Konsole aus.

```
XXXXXXXXXXXXXXXXXX
X X XXXXXXXXXXXX*X
X$X XX      XXX.X
X.X XX XXX XXX. X
X...XX XXX XXX. X
XXX.X...XX XXX. X
XXX...X.....X
XXXXXXXXXXXXXXXXXX
```

Aufgabe 2. Brainfuck ist eine sogenannte esoterische Programmiersprache: Das sind Sprachen, die meist zu wissenschaftlichen oder theoretischen Zwecken, oder einfach zum Spaß entwickelt wurden.



Brainfuck besteht nur aus 8 Befehlen: > < + - , . [] – alle anderen Zeichen werden als Kommentar interpretiert. Diese Befehle werden, wie bei C auch, nacheinander ausgeführt. Sie operieren auf einem (potentiell unendlich langen) Speicherband (welches aus Zellen besteht in denen jeweils ein `char` steht) indem sie einen Lese-/Schreibkopf über das Band bewegen und Zeichen lesen/schreiben lassen. Das Band ist überall mit 0 vorinitialisiert und der Lese-/Schreibkopf startet an “Position 0” des Bandes. Die Befehle haben folgende Bedeutung:

>	schiebt den Lese-/Schreibkopf um eine Position nach rechts
<	schiebt den Lese-/Schreibkopf um eine Position nach links
+	inkrementiert den Bandwert unter dem Lese-/Schreibkopf um 1
-	dekrementiert den Bandwert unter dem Lese-/Schreibkopf um 1
.	gibt den Wert unter dem Lese-/Schreibkopf aus
,	liest ein Zeichen vom Benutzer ein und schreibt es unter den Lese-/Schreibkopf
[springt zum zugehörigen]-Befehl, wenn der Wert unter dem Lese-/Schreibkopf gleich 0 ist, sonst soll nichts passieren
]	springt zum zugehörigen [-Befehl, wenn der Wert unter dem Lese-/Schreibkopf verschieden von 0 ist

So sieht ein “Hallo-Welt”-Programm in Brainfuck aus:

```

1 ++++++
2 [ >++++++>++++++>++++>+<<<<- ]
3 >++.
4 >+.
5 ++++++.
6 +++.>+.
7 <<<<+++++++.
8 >+.+++
9 _____
10 >+.>.
```

Deine Aufgabe ist es nun, ein Programm zu schreiben, welches Brainfuck-Programme einlesen und ausführen kann.