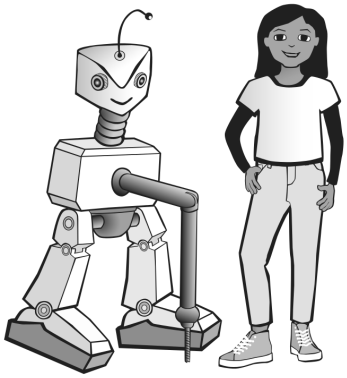
© Kinderlabor, 2022

Informatik, fertig, los!

Arbeitsheft

Marsroboter
Robo



Programmiererin
Rana

Der Roboter Robo wurde mit einem Raumschiff auf eine lange Reise geschickt, um unseren Nachbarplaneten Mars zu untersuchen. Nun ist Robo sicher auf dem Mars angekommen und hat erste Fotos von seinem Landeplatz geschickt! Die Programmiererin Rana steuert Robo über Funk von der Erde aus und sagt ihm in seiner Sprache, was er tun soll. Erst untersucht Robo an seinem Landeplatz den Marsboden, dann erforscht er die nähere Umgebung. Am Ende geht Robo auf Wanderschaft über den Mars.

Du schlüpfst in die Rolle von Rana. Zuerst lernst du die Sprache kennen, die Robo versteht. Dann benutzt du diese Sprache, um Robo zu steuern, damit er seine Aufgaben schnell und fehlerfrei erledigen kann.

Dabei lernst du am Beispiel von Robo, wie du einen Computer programmierst, und dieses Wissen ist auch auf der Erde sehr nützlich!

Erste Schritte: Seite 3

Robo hat in seinem Kopf einen Computer und kein Gehirn. Er versteht unsere Sprache nicht, hat aber wie alle Computer eine eigene Sprache. In der kannst du ihm sagen, was er machen soll. So eine Sprache heisst **Programmiersprache**. Sie besteht aus verschiedenen **Befehlen**, die du zu **Programmen** zusammensetzen kannst. Zuerst lernst du die Programmiersprache von Robo kennen.

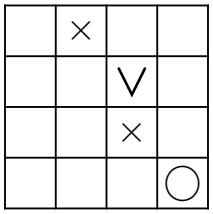
Programmieren: Seite 14

Programmieren heisst, einem Computer oder Roboter Befehle zu geben, damit er genau das macht, was du von ihm willst. Welche Befehle Robo versteht, hast du schon gelernt. Jetzt benutzt du diese Befehle, um Robo schnell und sicher ans Ziel seiner jeweiligen Aufgabe zu bringen! Dabei musst du Robo manchmal um Hindernisse herumsteuern. Du lernst auch, was du machen kannst, wenn Robo technische Probleme hat und bestimmte Befehle gerade nicht ausführen kann.

Wiederholungen: Seite 22

Du hast gelernt, Robo ans Ziel zu steuern, damit er dort den Marsboden untersuchen kann. Jetzt soll Robo für uns ein vorgegebenes Gebiet erforschen, indem er alle Felder im Gebiet besucht. Bei einer solchen Forschungsaufgabe macht Robo oft immer wieder das Gleiche. So wie du es auch machst, wenn du zum Beispiel die Bäume im Park zählst. Du lernst, wie du dir die Programmierung von Forschungsaufgaben erleichtern kannst, indem du **Wiederholungsbefehle** benuzt. Mit solchen Befehlen kannst du Robo sagen, dass er etwas mehrmals machen soll.

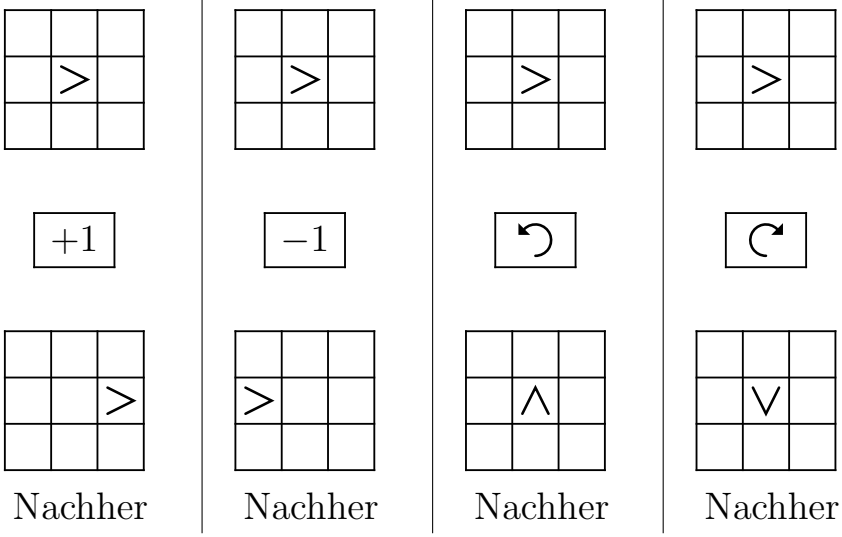
Eine Aufgabe für Robo sieht so wie im Bild rechts aus: Hier schaut Robo nach unten (⋁). Er kann aber auch nach rechts (>), oben (⋀) oder links (<) schauen. Robo soll am Ziel (◯) den Marsboden untersuchen und muss dafür zum Ziel gesteuert werden. Felder mit Hindernissen (×) darf Robo nicht betreten.



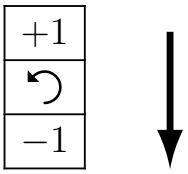
Robo versteht vier verschiedene Befehle. Diese bilden die Programmiersprache, in der du ihm sagen kannst, was er machen soll:

- [+1] : Gehe ein Feld vorwärts!
- [↺] : Mache eine Vierteldrehung gegen den Uhrzeigersinn!
- [−1] : Gehe ein Feld rückwärts!
- [↻] : Mache eine Vierteldrehung im Uhrzeigersinn!

Die Befehle [+1] und [−1] sind die Laufbefehle, [↺] und [↻] sind die Drehungsbefehle. Hier siehst du am Beispiel, was jeweils passiert, wenn Robo einen der vier Befehle ausführt:



Wenn du Robo ans Ziel steuern willst, musst du ihm meistens mehrere Befehle geben. Ein Stapel von Befehlen ist ein Programm und sieht zum Beispiel so aus:

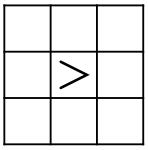


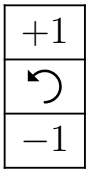
Programme werden **von oben nach unten** gelesen. Mit diesem Programm gibst du Robo also folgende Befehle:

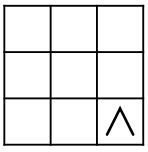
Gehe ein Feld vorwärts!
Mache eine Vierteldrehung gegen den Uhrzeigersinn!
Gehe ein Feld rückwärts!

Hier siehst du, welche Wirkung dieses Programm hat:

Vorher

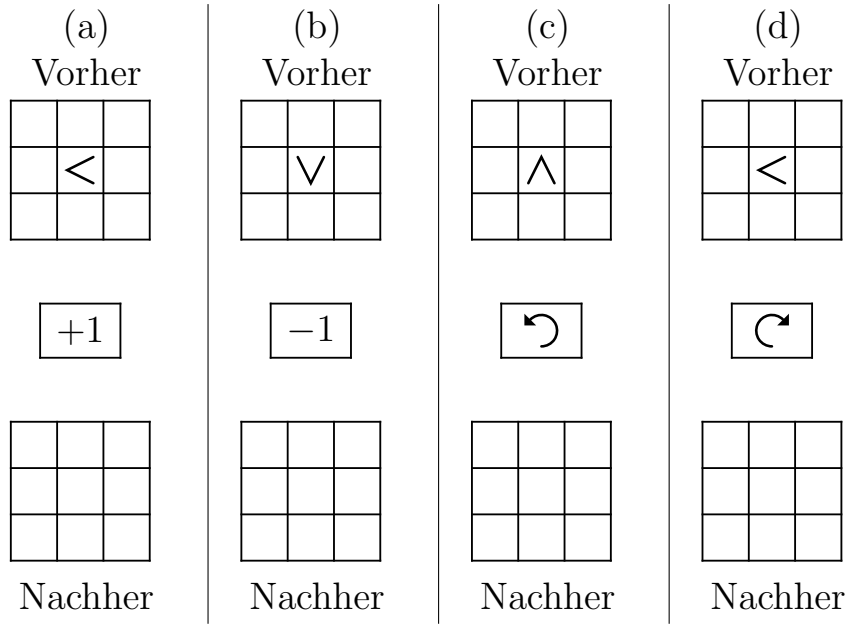




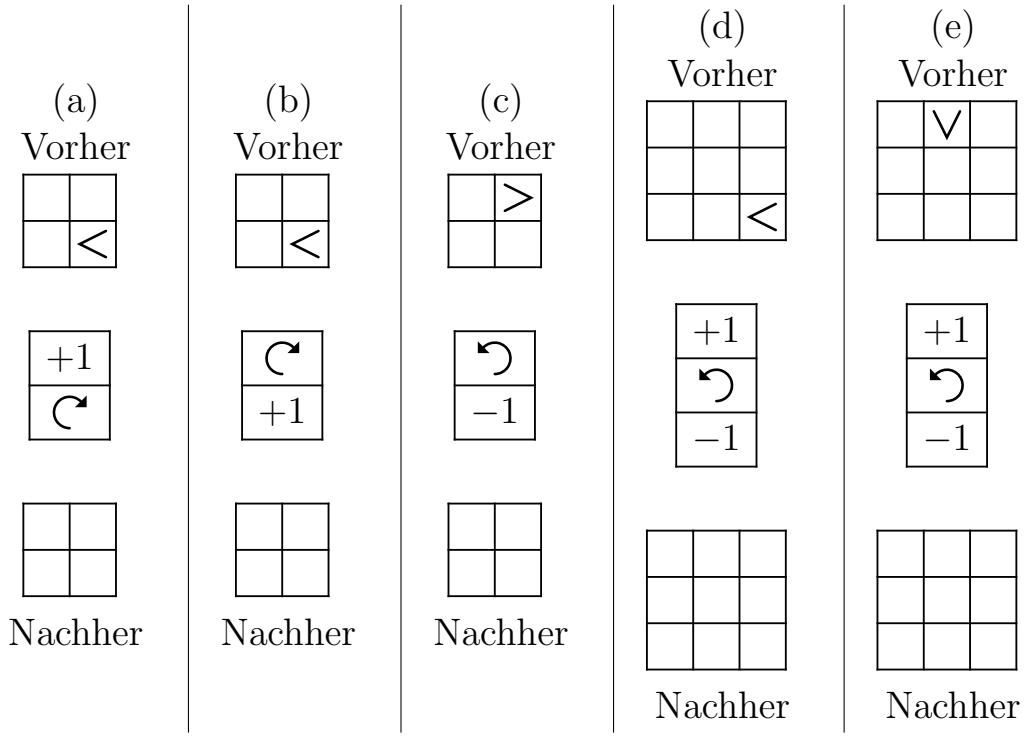


Nachher

Aufgabe 1. Zeichne jeweils ein, wo Robo nach der Ausführung des Befehls ist, und in welche Richtung er schaut!

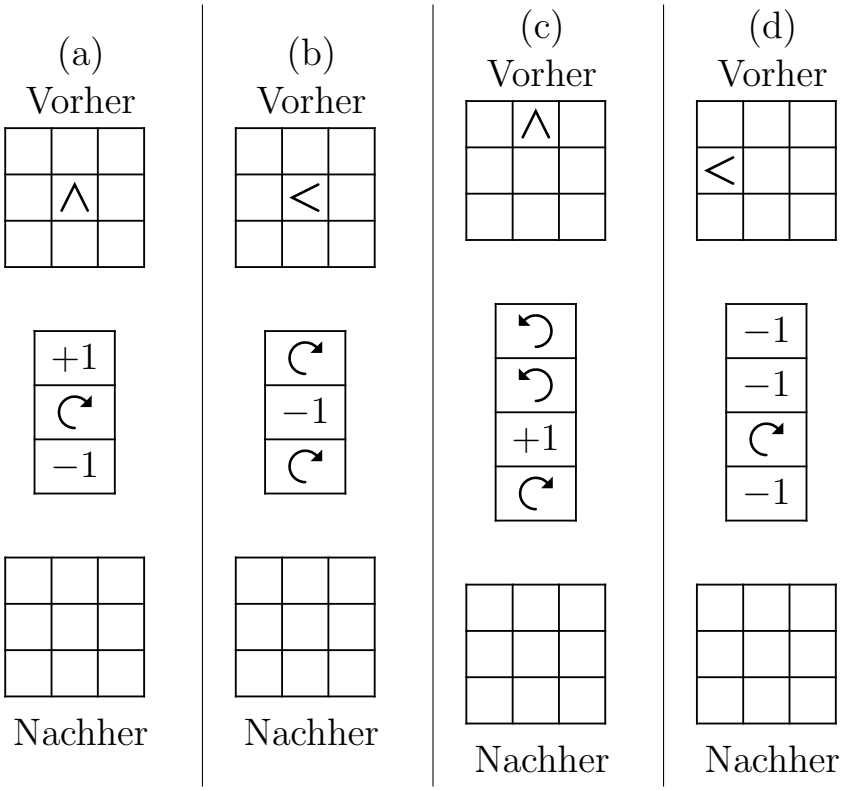


Aufgabe 2. Zeichne jeweils ein, wo Robo nach der Ausführung des Programms ist, und in welche Richtung er schaut!



Finde heraus, wie Robo funktioniert!

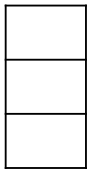
Aufgabe 3. Zeichne jeweils ein, wo Robo nach der Ausführung des Programms ist, und in welche Richtung er schaut!



Aufgabe 4. In Aufgabe 3 ist Robo in einem Fall unnötig kompliziert gelaufen. Finde die Teilaufgabe, bei der du Robo mit weniger Befehlen von Vorher zu Nachher bringen kannst! Schreibe den Buchstaben der Teilaufgabe und dein kürzeres Programm auf!

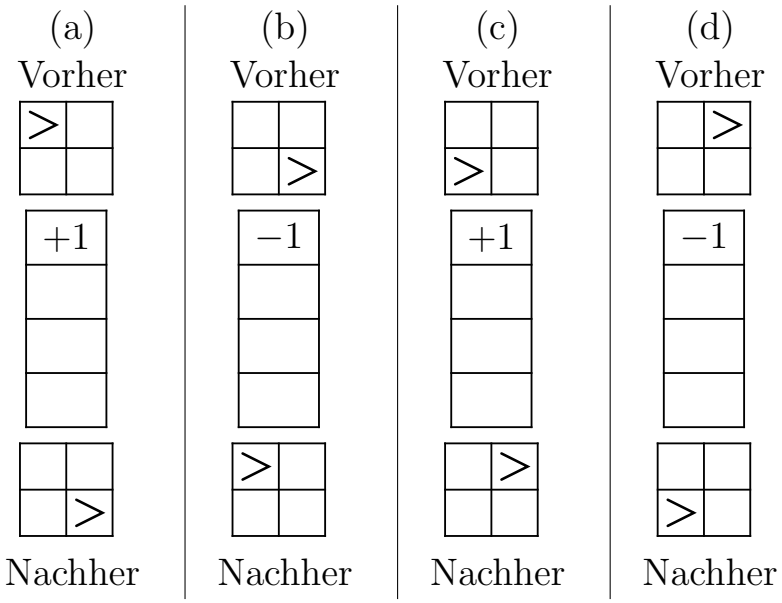
Es ist Teilaufgabe:

Mein kürzeres Programm sieht so aus:

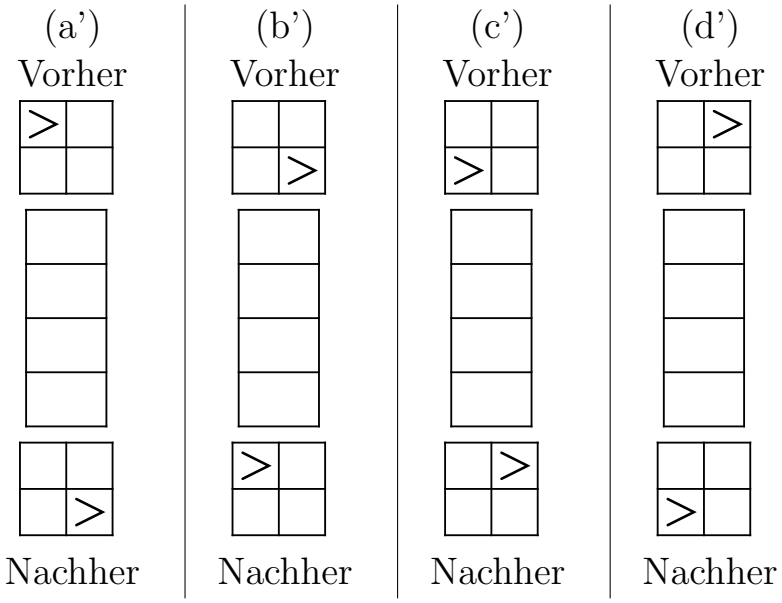


Wenn du das lösen kannst, kennst du dich schon gut mit Robo aus!

Aufgabe 5. Robo soll getestet werden. Wohin er gehen soll, ist dabei schon vorgegeben. Jedes Testprogramm soll immer alle vier Befehle enthalten! Trage die fehlenden Befehle ein!



Findest du noch andere Lösungen mit allen vier Befehlen?



Diese Aufgabe ist auch ein echter Test für dich!

Aufgabe 6. Robo im Labyrinth: In welchen dieser Gitter kann er ans Ziel kommen, ohne das Gitter zu verlassen oder in ein Hindernis (×) zu laufen? Wenn es einen Weg für Robo gibt, zeichne ihn ein!

(a)

∧	×				
	×		×		
			×	×	
		×			
		×		×	
	×	○			

(b)

∧	×				
×					
					○

(c)

×	>		×		
○	×			×	
		×			×
×			×		
	×			×	
		×			

(d)

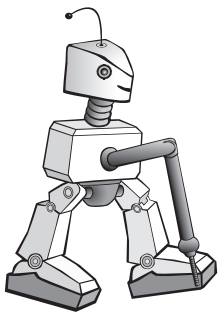
○	×				
	×			×	
	×			×	
	×			×	
	×			×	
				×	>

(e)

×					
○	×				
×					
					<

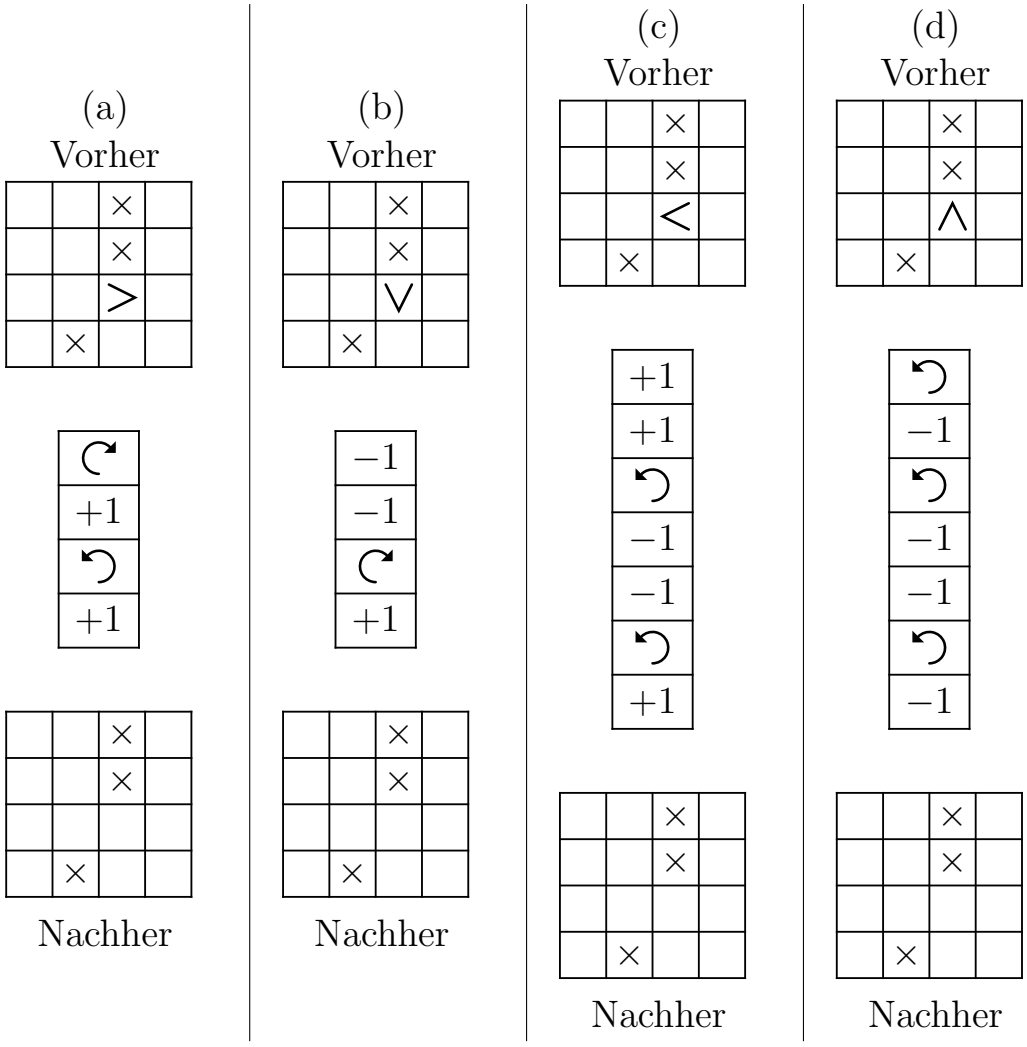
(f)

		○			
×		×		×	
	×		×		×
			∧		



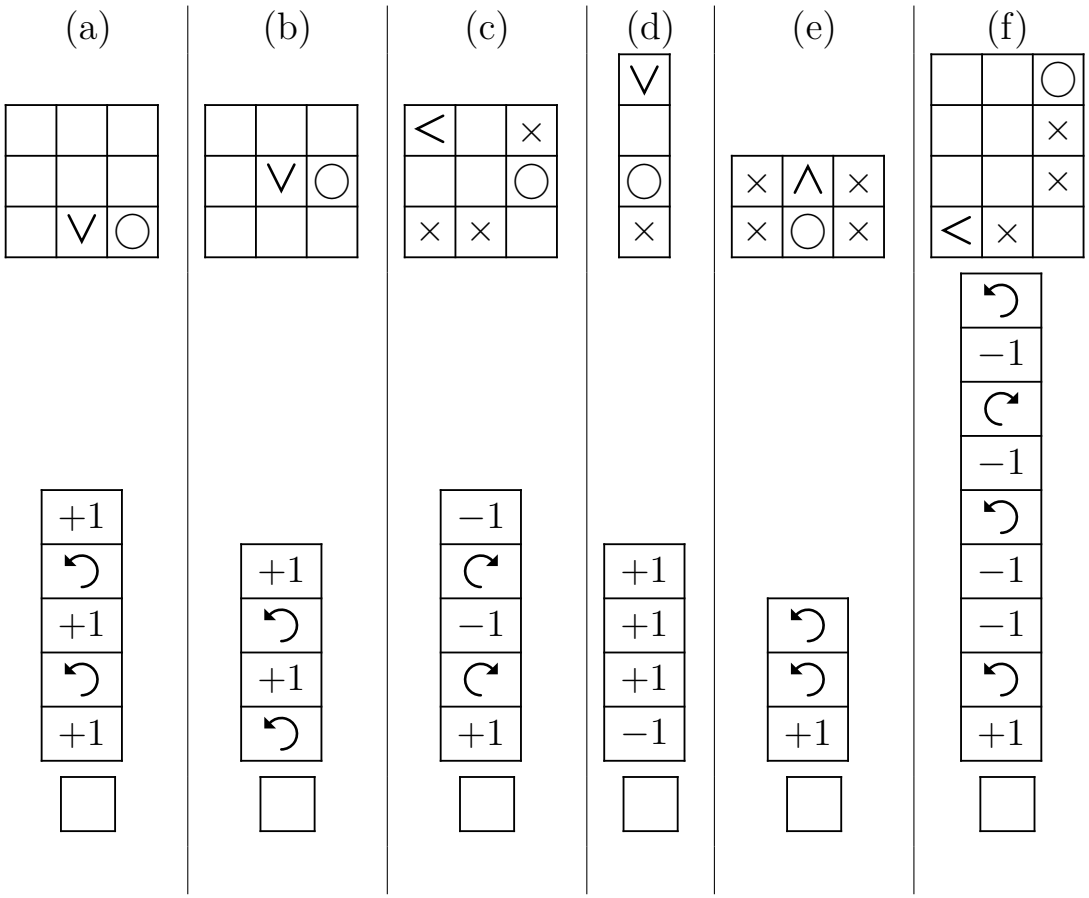
Zeige, dass du genau verstehst, was Robo kann!

Aufgabe 7. Du willst nicht, dass Robo in ein Hindernis läuft und beschädigt wird! Deshalb musst du Programme erkennen, bei denen das passiert. Läuft Robo mit den Programmen unten in ein Hindernis? Wenn ja, umkreise das Hindernis, in das Robo zuerst läuft! Wenn nein, zeichne ein, wo Robo nach der Ausführung des Programms ist, und in welche Richtung er schaut!



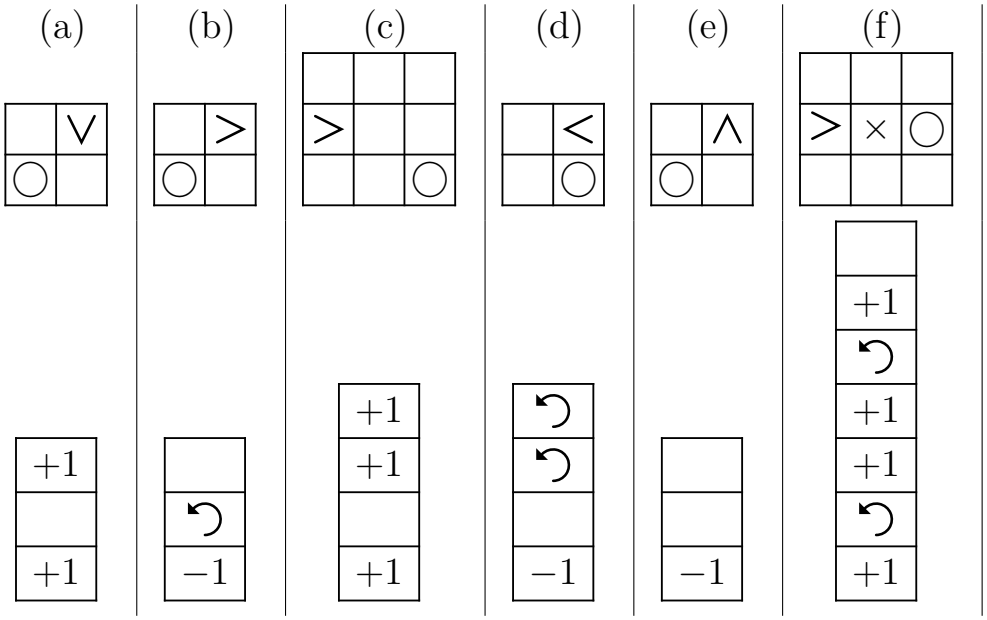
Robo freut sich sehr, wenn du gut auf ihn aufpasst!

Aufgabe 8. Einige der Programme unten bringen Robo ans Ziel, einige nicht. Setze ein Kreuz in das Lösungsfeld unter die Programme, die richtig sind! Aufgepasst: Robo darf weder über Hindernisse laufen noch das Gitter verlassen!

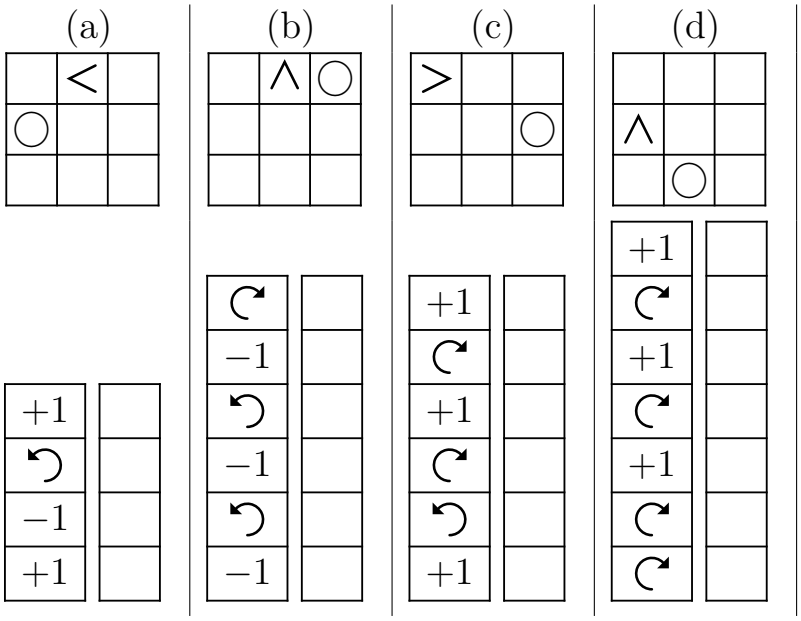


Aufgabe 9. Robo ist auf einem Startfeld und führt dann ein Programm mit 7 Laufbefehlen (+1, −1) und einigen Drehungsbefehlen (↺, ↻) aus. Kann es sein, dass Robo nach der Ausführung des Programms wieder auf dem Startfeld ist? Erkläre deine Antwort!

Aufgabe 10. In jedem der Programme unten wurden aus Versehen Befehle ausradiert. Welche Befehle müssen dort jeweils stehen, damit Robo ans Ziel kommt? Ergänze die richtigen Befehle!

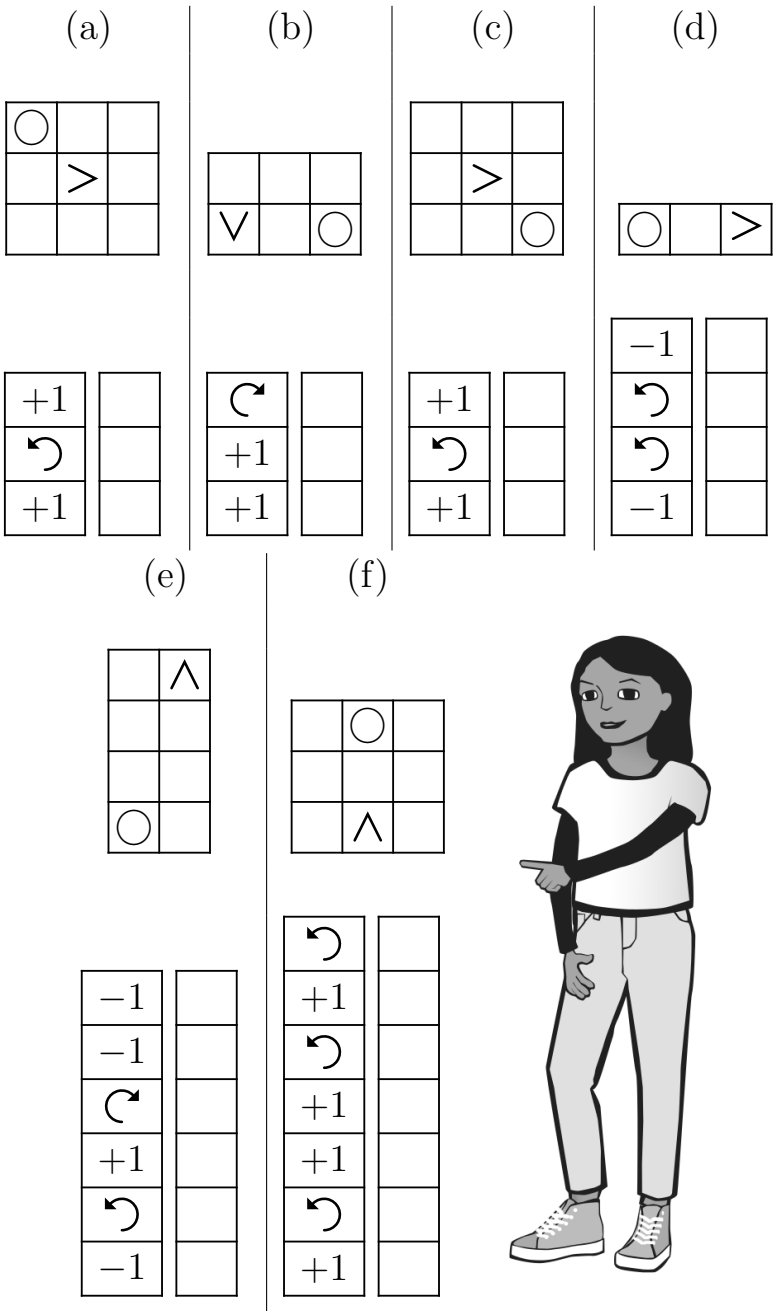


Aufgabe 11. Jedes der Programme unten enthält einen Befehl zu viel. Markiere den Befehl, den du herausnehmen musst, damit Robo ans Ziel kommt! Mache dazu ein Kreuz ins Kästchen neben dem Befehl.



Robo ist dir dankbar, wenn du Fehler in seinen Programmen findest!

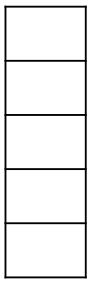
Aufgabe 12. In jedem der Programme unten gibt es einen Fehler, so dass Robo nicht ans Ziel kommt: An einer Stelle wurde [↻] mit [↺] verwechselt, oder [+1] mit [−1]. Mache ein Kreuz neben der Stelle, an der die Verwechslung stattgefunden hat!



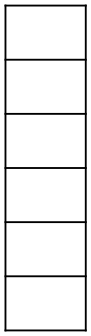
Rana freut sich übrigens auch sehr über deine Hilfe!

Aufgabe 13. Eine Wissenschaftlerin, die Robos Programmiersprache nicht kennt, braucht deine Hilfe. Sie sagt dir in normaler Sprache, was Robo machen soll. Dann bittet sie dich, Robo so zu programmieren, dass er genau das macht. Schreibe Programme dafür, die Robo versteht!

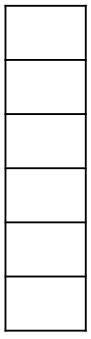
(a) Gehe zwei Felder vorwärts und mache eine Vierteldrehung im Uhrzeigersinn; dann gehe wieder zwei Felder vorwärts!



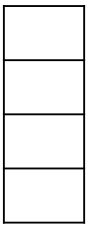
(b) Mache eine Vierteldrehung gegen den Uhrzeigersinn und gehe ein Feld rückwärts. Dann mache das gleiche noch einmal! Danach gehe noch zwei Felder rückwärts!



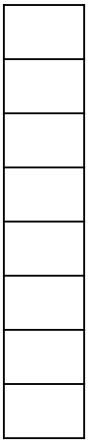
(c) Gehe drei Felder vorwärts, und dann rückwärts, bis du wieder an deinem ursprünglichen Platz bist!



(d) Mache so lange Vierteldrehungen im Uhrzeigersinn, bis du wieder in die gleiche Richtung wie am Anfang schaust!



(e) Mache einen kleinen Rundgang: ein Feld vorwärts, Vierteldrehung gegen den Uhgrzeigersinn; ein Feld vorwärts, Vierteldrehung gegen den Uhrzeigersinn; und so weiter, bis du wieder am Ausgangspunkt bist!

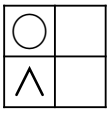


Aufgabe 14. Arbeit zu zweit: Schreibe ein kurzes Programm auf (höchstens 6 Befehle), ohne dass dein Gegenüber es sieht. Du kannst es zum Beispiel hinter deinem Heft verstecken. Es kommt nicht darauf an, dass es ein sinnvolles Programm ist, du kannst auch einen "zufälligen" Stapel aus 6 Befehlen machen.

Dann stellst du deinen Robo auf dein Gitter und zeigst deinem Gegenüber Schritt für Schritt, wie Robo dein Programm ausführt! Dein Gegenüber soll beim Beobachten dein Programm erraten und es selber aufschreiben. Am Ende vergleicht ihr eure beiden Programme. Wenn sie nicht gleich sind, versucht herauszufinden, warum Fehler passiert sind! Dann tauscht ihr.

Aufgabe 15. Schreibe für jedes der Gitter ein Programm, mit dem Robo ans Ziel kommt! Schreibe das Programm unter dem Gitter auf! Bei (g)–(k) musst du auch auf Hindernisse achten!

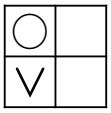
(a)



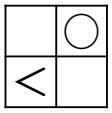
(b)



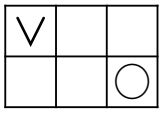
(c)



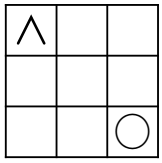
(d)



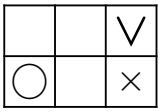
(e)



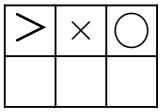
(f)



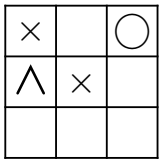
(g)



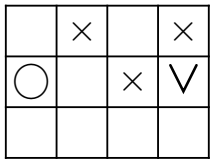
(h)



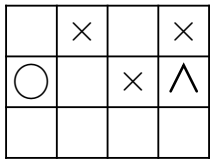
(i)



(j)

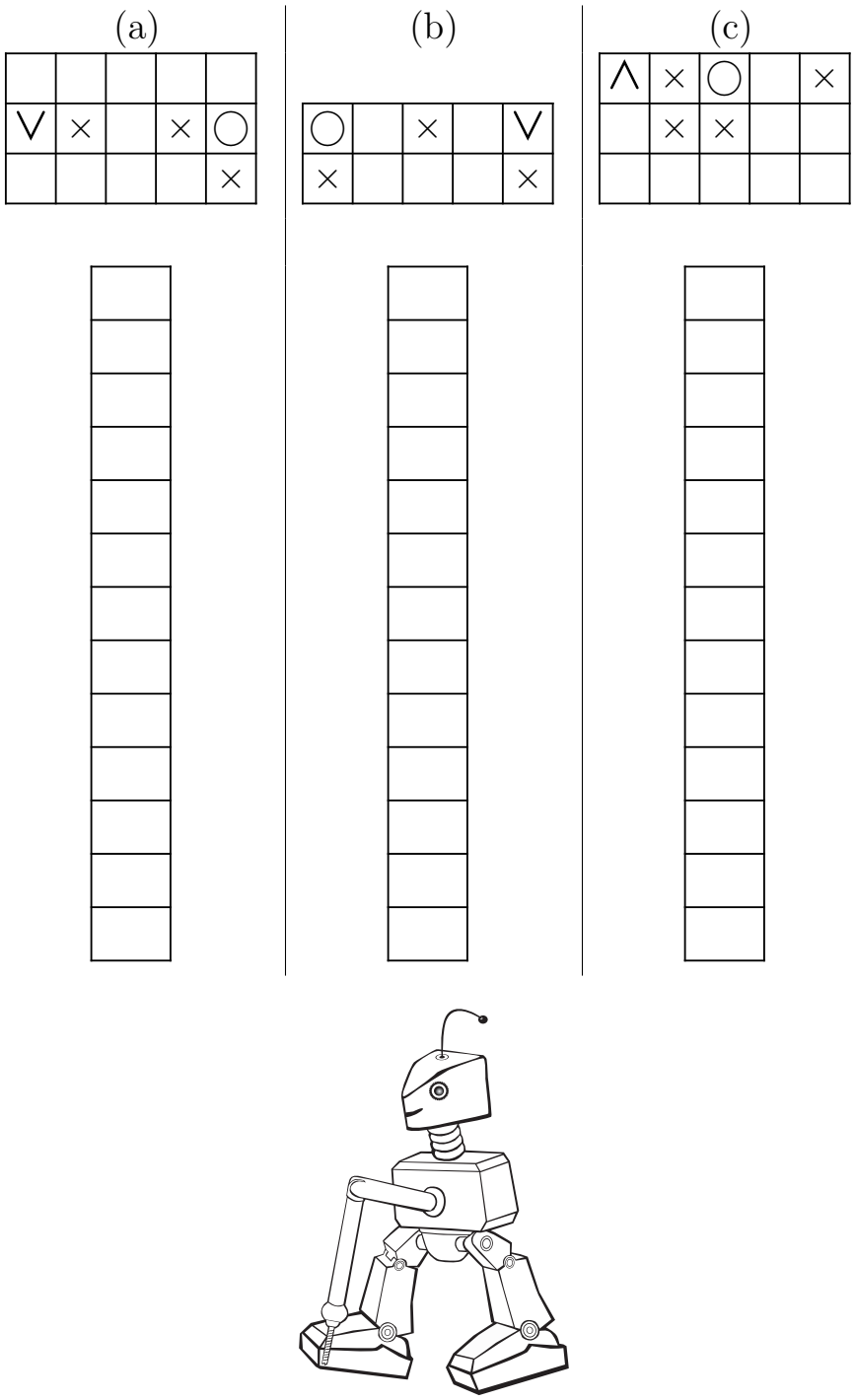


(k)



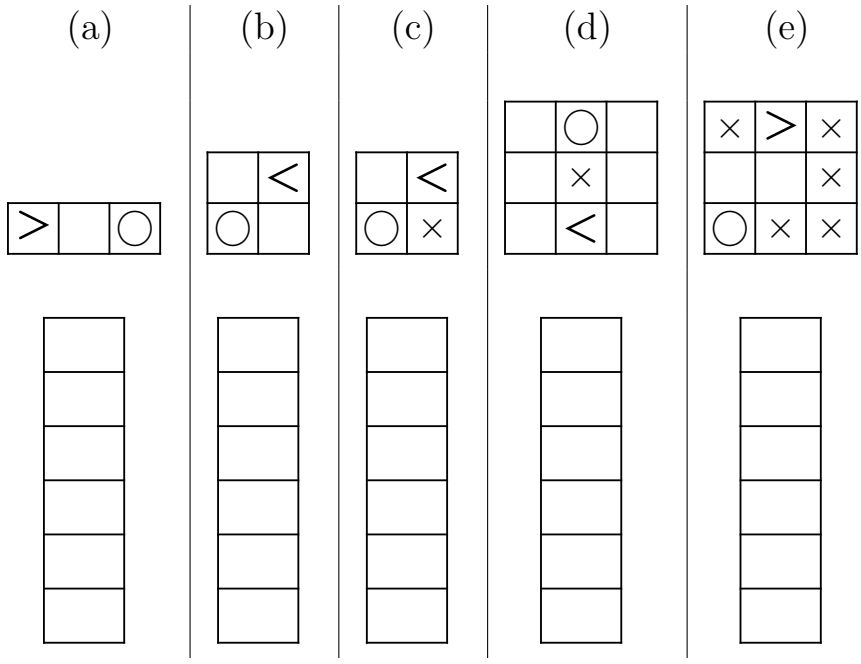
Jetzt geht es richtig mit dem Programmieren los!

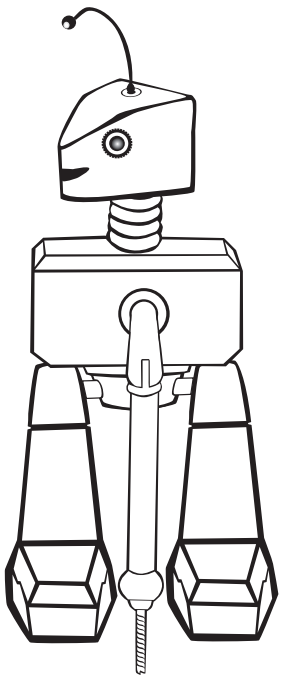
Aufgabe 16. Schreibe für jedes der Gitter ein Programm, mit dem Robo ans Ziel kommt! Wie vorher darf Robo Felder mit Hindernisssen nicht betreten.



Hier wird es schon ganz schön knifflig! Viel Erfolg!

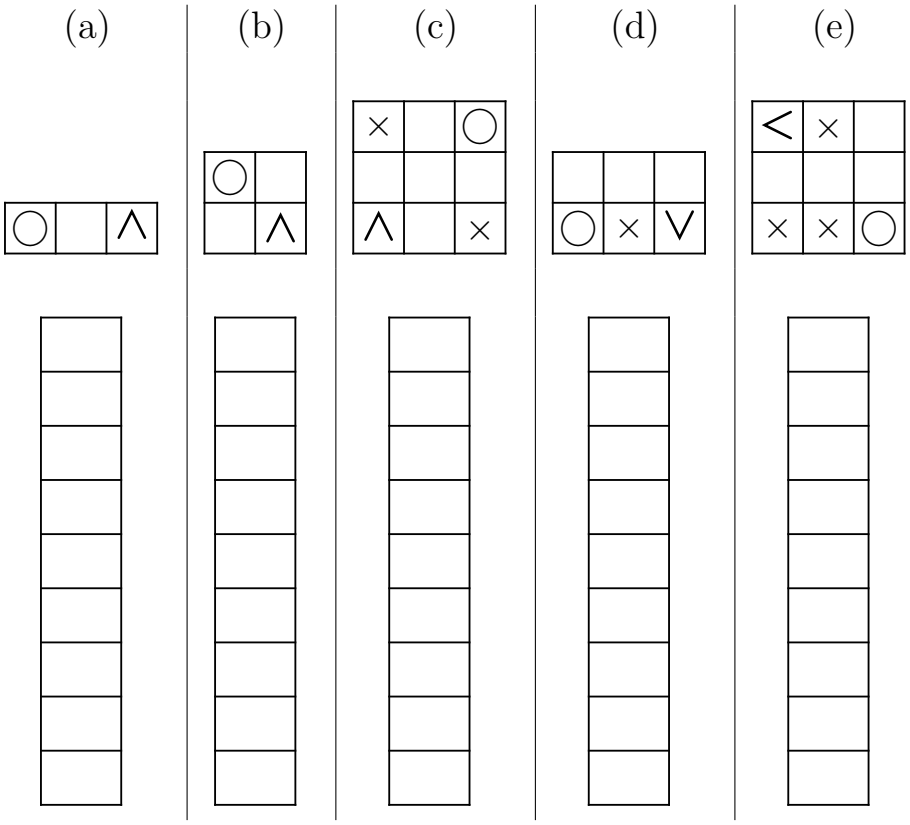
Aufgabe 17. Bei Robo streikt der Vorwärtsgang. Eine Technikerin versucht, das Problem zu lösen, aber momentan kann Robo nur noch rückwärts laufen. Bringe Robo ans Ziel, ohne den Befehl $\boxed{+1}$ zu benutzen!

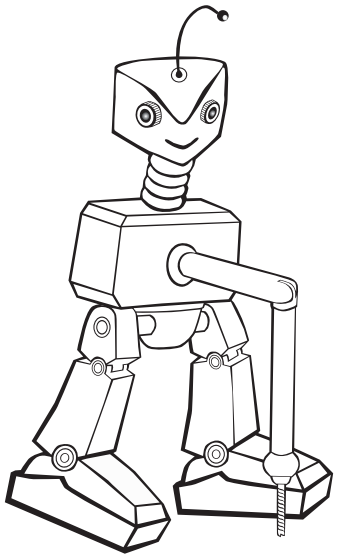




Hoffentlich kann Robo bald wieder vorwärts laufen!

Aufgabe 18. Bei Robo funktioniert der Vorwärtsgang zwar wieder, dafür ist jetzt die Lenkung gestört. Robo kann sich nur noch im Uhrzeigersinn drehen. Löse die folgenden Aufgaben, ohne den Befehl ↻ zu benutzen!

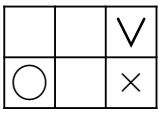




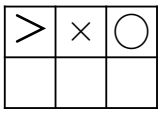
Schon wieder ein Problem! Rana braucht deine Hilfe, um Robo zu lenken!

Aufgabe 19. Bei Robo ist die Lenkung leider immer noch gestört. Mal funktioniert [↺] nicht, mal streikt [↻]. Löse die Gitter (g)–(k) aus Aufgabe 15 noch einmal, ohne den jeweils angegebenen Befehl zu benutzen! Brauchst du dafür mehr Befehle als in Aufgabe 15?

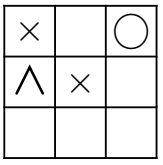
(a) kein [↺] !



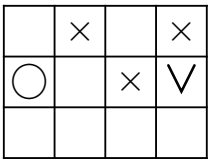
(b) kein [↺]



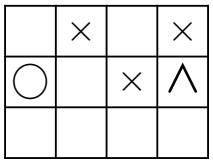
(c) kein [↺] !

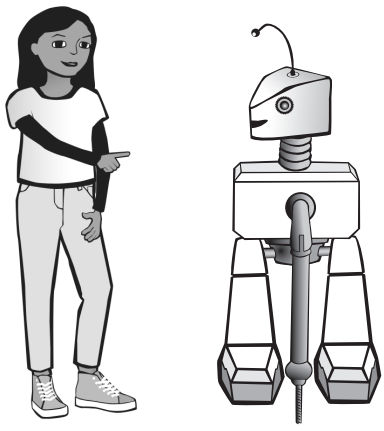


(d) kein [↻] !



(e) kein [↻] !

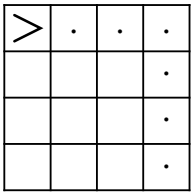




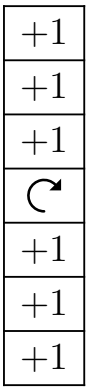
Du hast ja schon Erfahrungen damit, Lenkungsprobleme zu lösen! Viel Erfolg!

Aufgabe 20. ▁▃ Erkläre, wie Robo immer noch jede Aufgabe lösen kann, wenn der Befehl [↻] nicht mehr funktioniert! Wenn du ein normales Programm hast (alle vier Befehle sind erlaubt), wie verwandelst du es in ein Programm, das die gleiche Aufgabe löst, aber den Befehl [↻] nicht benutzt?

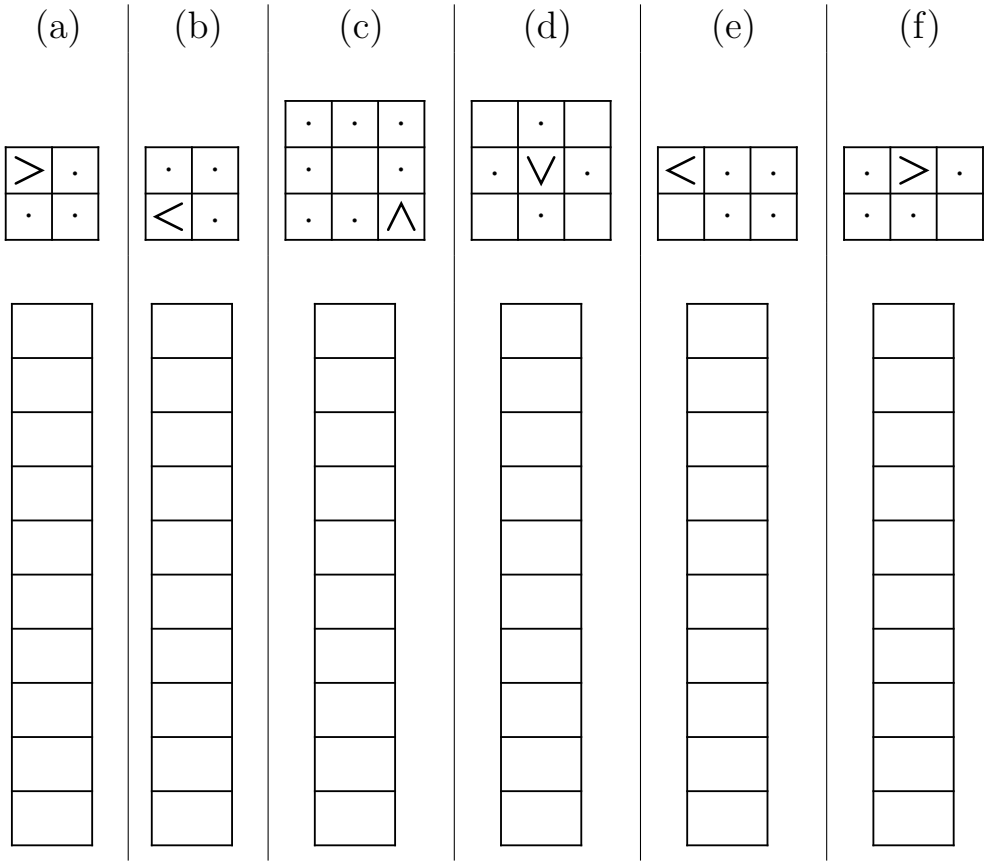
Aufgabe 21. ▃▆ Sogar wenn die *beiden* Befehle [+1] und [↻] nicht mehr funktionieren, kann Robo mit deiner Hilfe immer noch alle Aufgaben lösen! Wenn du ein normales Programm hast (alle vier Befehle sind erlaubt), wie verwandelst du es in ein Programm, das die gleiche Aufgabe löst, aber die Befehle [+1] und [↻] nicht benutzt?



Eine Forschungsaufgabe für Robo sieht so wie im Bild links aus. Robo soll auf seinem Weg alle markierten Felder (·) besuchen, aber keine anderen. Er darf dabei ein markiertes Feld (und auch das Startfeld) mehrmals besuchen. In welcher Reihenfolge Robo die markierten Felder besucht, spielt keine Rolle. Rechts siehst du eine Lösung für diese Forschungsaufgabe.

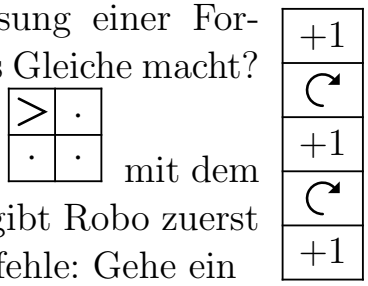


Aufgabe 22. Schreibe Programme für Robo mit denen er die folgenden Forschungsaufgaben lösen kann!



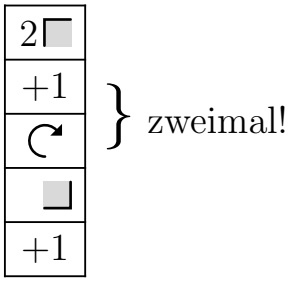
Jetzt gibt es noch einmal ganz neue Aufgaben für Robo. Viel Erfolg!

Ist dir aufgefallen, dass Robo bei der Lösung einer Forschungsaufgabe mehrmals hintereinander das Gleiche macht?



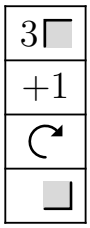
Hast du zum Beispiel die Forschungsaufgabe mit dem Programm rechts gelöst? Dieses Programm gibt Robo zuerst zweimal hintereinander die gleichen zwei Befehle: Gehe ein Feld vorwärts und mache eine Vierteldrehung im Uhrzeigersinn! Am Ende macht Robo noch einen Schrit vorwärts.

Das können wir auch so schreiben:



} zweimal!

Trick: Wenn wir Robo am Ende noch einmal drehen, hat er zwar mehr Arbeit bei der Forschungsaufgabe, wir dafür aber weniger:



} dreimal!

Das sind Wiederholungen. Um Wiederholungen machen zu können, erweitern wir die Programmiersprache von Robo um vier neue Wiederholungsbefehle.

 : Führe den Stapel bis zum  zweimal aus!

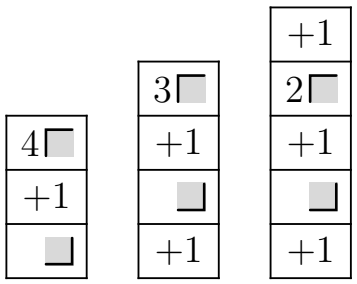
 : Führe den Stapel bis zum  dreimal aus!

 : Führe den Stapel bis zum  viermal aus!

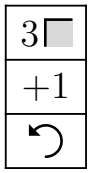
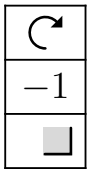
 : Ende der Wiederholung!

Die Wiederholung geht vom Anfang (,  oder ) bis zum Ende (). Was davor oder danach steht, gehört nicht mehr zur Wiederholung und wird normal vorher oder nachher ausgeführt.

Mit allen diesen Programmen geht Robo zum Beispiel vier Felder vorwärts:



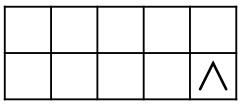
Eine Wiederholung hat immer Anfang *und* Ende, sonst ergibt sich ein ungültiges Programm, mit dem Robo nichts anfangen kann, so wie in diesen Beispielen:

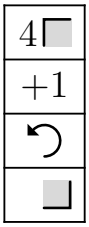
ungültig! →   ← ungültig!

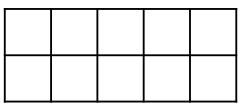
Aufgabe 23. Zeichne jeweils ein, wo Robo nach der Ausführung des Programms ist, und in welche Richtung er schaut!

(a)

Vorher





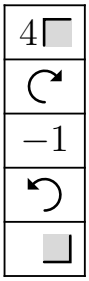


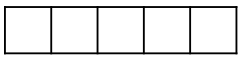
Nachher

(b)

Vorher







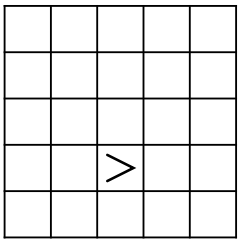
Nachher

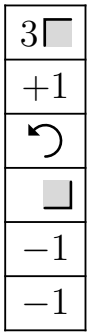
Das Arbeiten mit Wiederholungen ist nicht ganz einfach – aber du schaffst das!

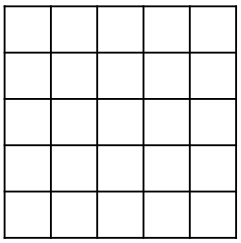
Aufgabe 24. Zeichne jeweils ein, wo Robo nach der Ausführung des Programms ist, und in welche Richtung er schaut!

(c)

Vorher



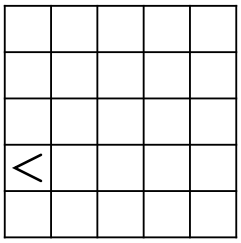


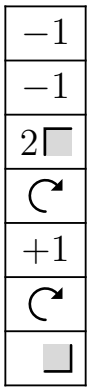


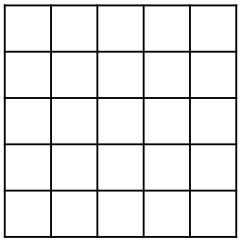
Nachher

(d)

Vorher



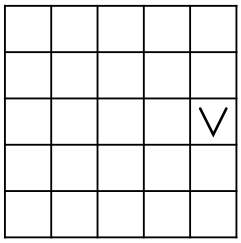


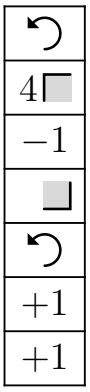


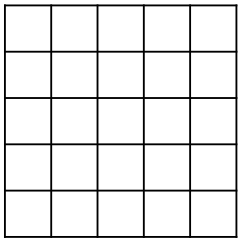
Nachher

(e)

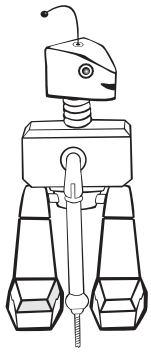
Vorher

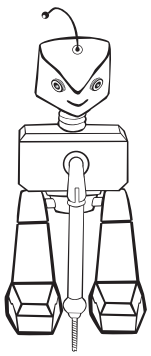


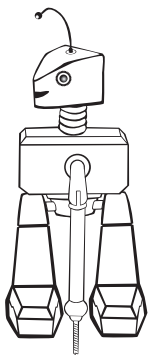




Nachher

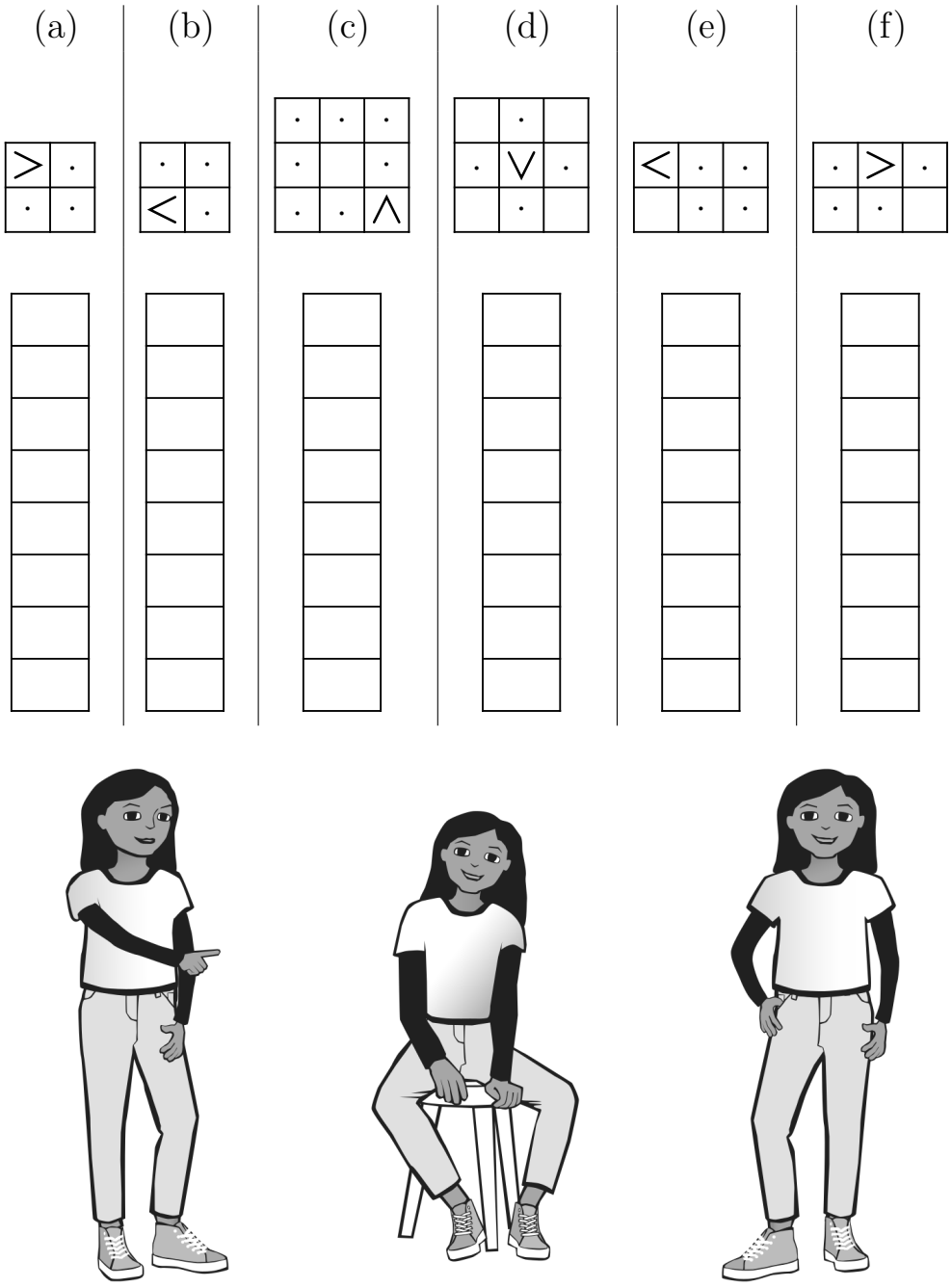






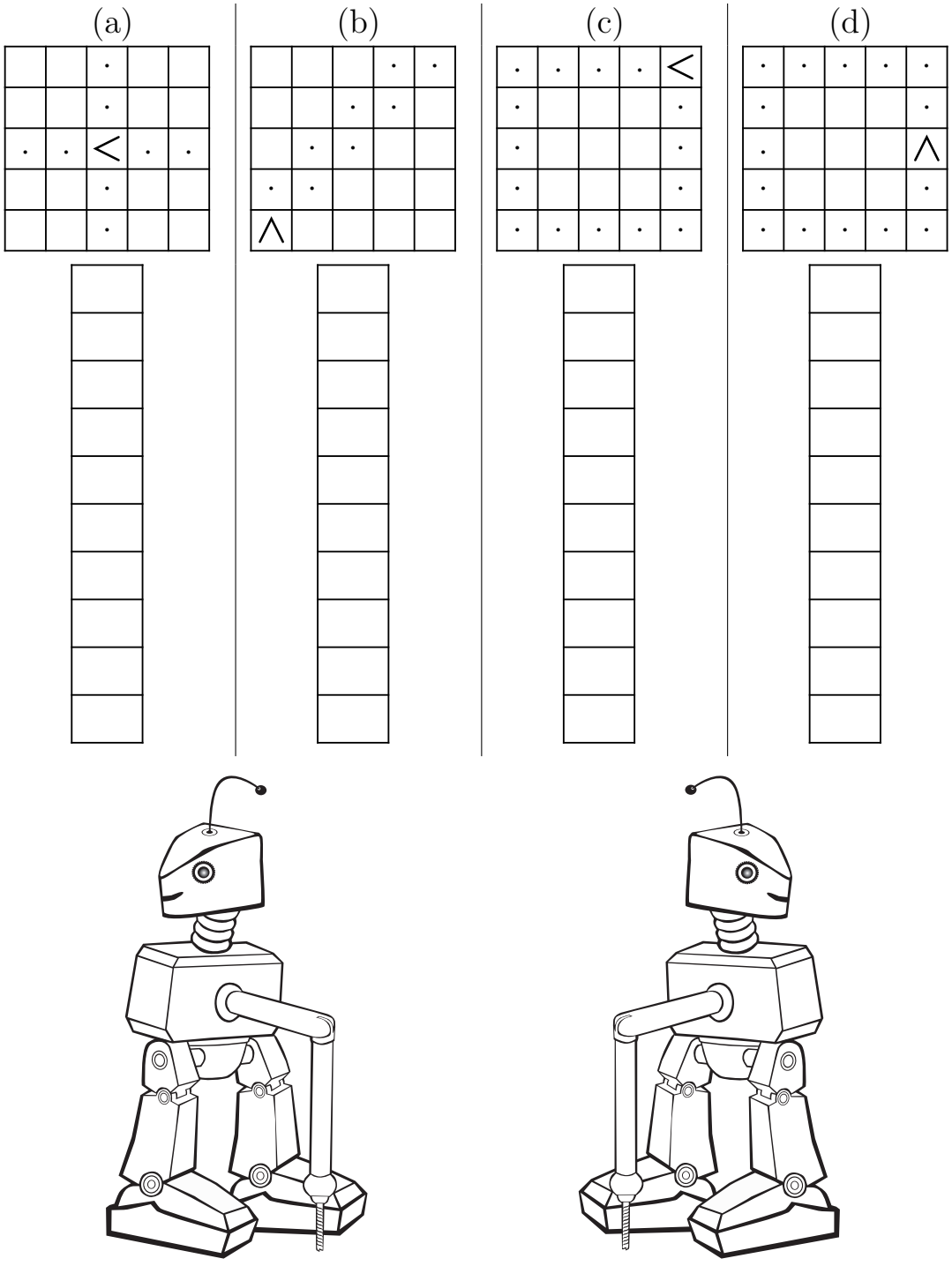
Robo mag Wiederholungen!

Aufgabe 25. Löse alle Forschungsaufgaben in Aufgabe 22 noch einmal mit Hilfe von Wiederholungen! Versuche dabei möglichst kurze Programme zu bauen!



Rana findet Wiederholungen ziemlich praktisch! Du auch?

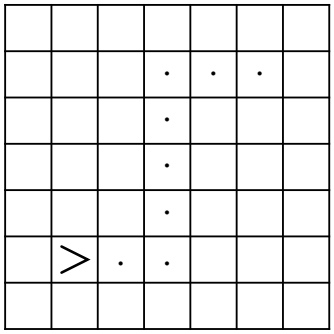
Aufgabe 26. Schreibe möglichst kurze Programme für Robo mit denen er die folgenden Forschungsaufgaben lösen kann! Benutze Wiederholungen, um Befehle zu sparen!



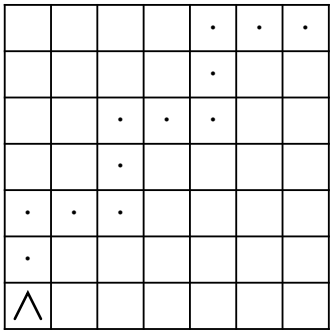
Bei dieser Aufgabe musst du schon einige Verrenkungen machen! Viel Erfolg!

Aufgabe 27. Schreibe Programme für Robo, mit denen er die Forschungsaufgaben unten mit der gegebenen Anzahl von Befehlen lösen kann! Du kannst in einem Programm auch mehrere Wiederholungen benutzen!

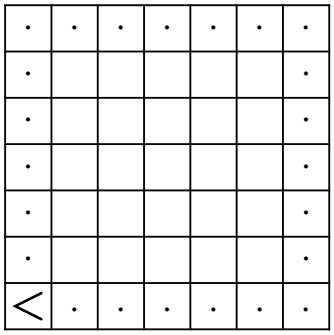
(a)



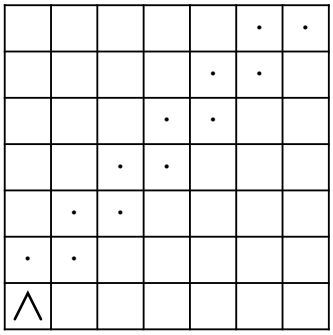
(b)

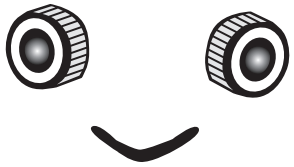


(c)



(d)

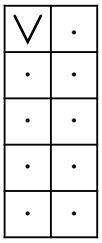


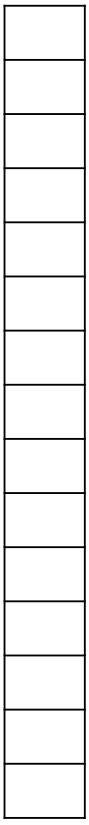


Achtung, jetzt wird es noch schwieriger! Schaffst du diese Aufgabe?

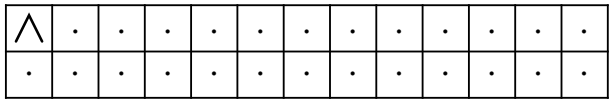
Aufgabe 28. Wie kannst du Robo jeweils das ganze Gitter erkunden lassen? Schreibe die kürzesten Programme, die du finden kannst!

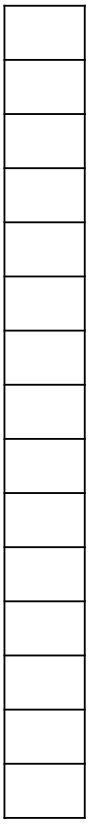
(a)



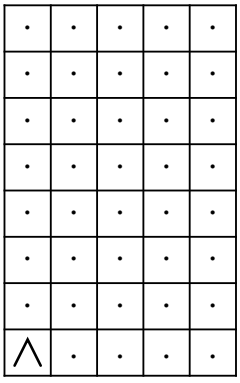


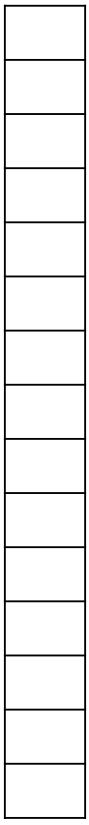
(b)





(c)





Das ist die letzte Aufgabe! Aber die hat es wirklich in sich! Viel Erfolg!!