

Jugendwettbewerb Informatik

1. Runde 2025

Klasse 5/6

Verbleibende Zeit: 0:29:38 (Lehrkräfte sind vom Zeitlimit ausgenommen.)

Quadrate



Chatbot



Weltraumaufräumer



Punktestand: 0 ★ / 12 ☆ (0 %)

Quadrature

VERSION ☆☆☆

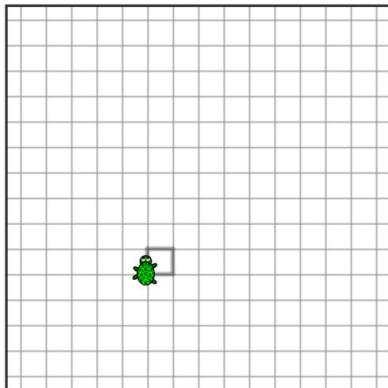
VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere die Schildkröte:

Die Schildkröte soll das Quadrat malen.



drehe nach rechts ▾

gehe  Schritte

0

wiederhole  10 mal:
mache

Schildkröten-Programm

Noch 12 von 12 Bausteinen verfügbar.



VERSION ☆☆☆

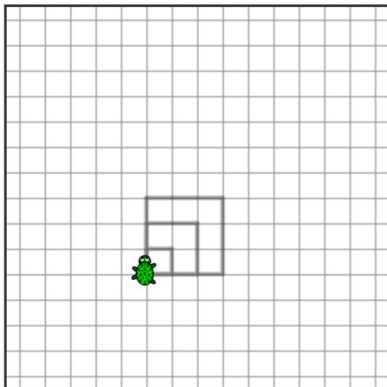
VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere die Schildkröte:

Die Schildkröte soll die Quadrate malen.



drehe nach rechts ↻

gehe ➡ Schritte

0

wiederhole 10 mal:

mache

zähle i von 1 bis 10 in Schritten von 1

mache



Schildkröten-Programm

Noch 16 von 16 Bausteinen verfügbar.



VERSION ☆☆☆

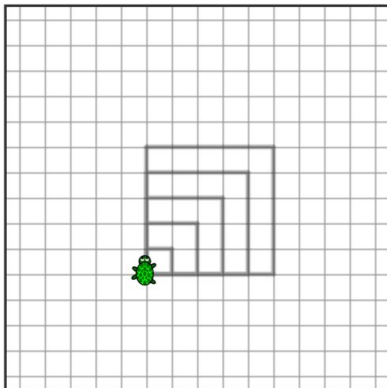
VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere die Schildkröte:

Die Schildkröte soll die Quadrate malen.



drehe nach rechts ↻

gehe ➡ Schritte

0

wiederhole 10 mal:

mache

zähle i von 1 bis 10 in Schritten von 1

mache



Schildkröten-Programm

Noch 10 von 10 Bausteinen verfügbar.



Chatbot

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere einen Chatbot:

Der Chatbot soll die Begrüßung in der Eingabe einlesen und in die Ausgabe schreiben.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

[+ WEITERE HINWEISE](#)

Eingabe:
Hal lo

Ausgabe:

schreibe

Programm

Noch 10 von 10 Bausteinen verfügbar.



lies Zeile

Eingabe

setze Eingabe auf

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere einen Chatbot:

Der Chatbot soll den Namen in der Eingabe einlesen und eine Begrüßung ausgeben. Dafür soll er **Hallo** vor den Namen schreiben.

Beispiel:

Eingabe:

Juna

Ausgabe:

Hallo Juna

[+ WEITERE HINWEISE](#)*Beachte: Dein Programm muss mit allen Testfällen*Eingabe:
Juna

Ausgabe:

schreibe

lies Zeile

“ ”

erstelle Text aus

Eingabe

setze Eingabe auf

Programm

Noch 10 von 10 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere einen Chatbot:

In der 1. Zeile der Eingabe steht ein Name. In der 2. Zeile steht eine Stimmung.

Zuerst soll der Chatbot eine Begrüßung ausgeben. Dafür soll er **Hallo** vor den Namen schreiben. Dann soll der Chatbot, falls die Stimmung **gut** ist,

Schön, dass es dir gut geht! ausgeben. Sonst soll er **Das verstehe ich nicht.** ausgeben. [WEITERE HINWEISE](#)

Beispiel:

Eingabe:
Juno
gut

Ausgabe:

schreibe

Programm

Noch 20 von 20 Bausteinen verfügbar.

lies Zeile

falls
mache
sonst

=

“ ”

erstelle Text aus

Eingabe

setze Eingabe auf

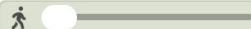
TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆

falls die Stimmung **gut** ist,

Schön, dass es dir gut geht!

 ausgeben. Sonst soll er

Das verstehe ich nicht.

 ausgeben.

Beispiel:

Eingabe:

Juna

gut

Ausgabe:

Hallo Juna

Schön, dass es dir gut geht!

VERSTECKTE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen

Eingabe:
Juno
gut

Ausgabe:

Aufgabenhinweise

Programmiere einen Chatbot:

In der 1. Zeile der Eingabe steht ein Name. In der 2. Zeile steht eine Stimmung.

Zuerst soll der Chatbot eine Begrüßung ausgeben. Dafür soll er **Hallo** vor den Namen schreiben. Dann soll der Chatbot, falls die Stimmung **gut** ist,

Schön, dass es dir gut geht!

 ausgeben. Sonst soll er **Das verstehe ich nicht.** ausgeben.

Beispiel:

Eingabe:

Juna

gut

Ausgabe:

Hallo Juna

Schön, dass es dir gut geht!

Weitere Hinweise:

Beim Zusammensetzen der Worte wird genau das zusammengesetzt, was im **“ ”** steht. Danke also daran, im Textbaustein nach dem **Hallo** ein Leerzeichen einzutragen.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf “ ”

TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆

falls die Stimmung **gut** ist,

Schön, dass es dir gut geht! ausgeben. Sonst soll er

Das verstehe ich nicht. ausgeben.

Beispiel:

Eingabe:

Juna

gut

Ausgabe:

Hallo Juna

Schön, dass es dir gut geht!

VERSTECKTE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

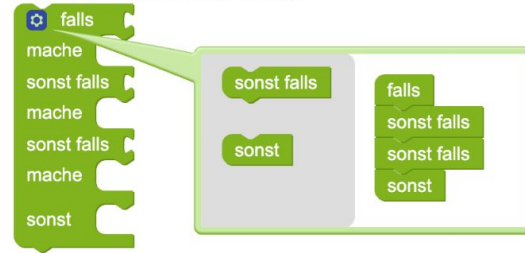
Eingabe:
Juno
gut

Ausgabe:

Aufgabenhinweise

Fallunterscheidung:

Du kannst der **falls mache** über das blaue Zahnrad weitere Fälle hinzufügen (oder entfernen), indem du sie vom linken grauen Kasten zum rechten weißen Kasten hinzufügst (oder aus dem weißen Kasten heraus ziehst):



Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Beste

Testfällen abzielt, zu vermeiden, stehen maximal vier **schreibe**-Bausteine zur

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



Weltraumaufräumer

VERSION ☆☆☆

VERSION ☆☆☆☆

VERSION ☆☆☆☆☆

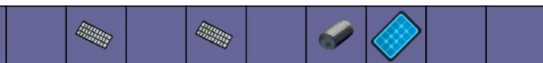


Programmiere den Roboter:

Der Roboter soll den Weltraum aufräumen. Dafür soll er allen Weltraumschrott ,  und  aufheben.

Hinweis: Der Roboter startet immer links. Das Gitter ist immer 10 Felder breit.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



hebe Weltraumschrott auf

gehe vorwärts

auf Weltraumschrott

falls
machewiederhole 10 mal:
mache

Roboter-Programm

Noch 10 von 10 Bausteinen verfügbar.



TEST 1

TEST 2



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆

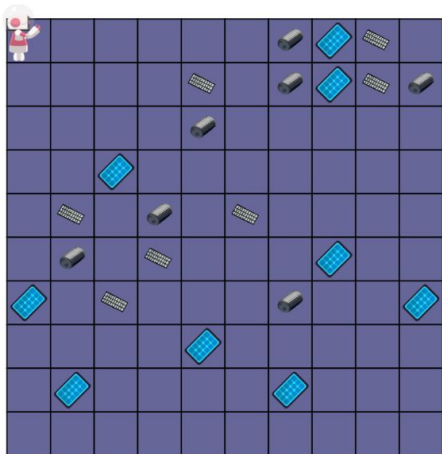


Programmiere den Roboter:

Der Roboter soll den Weltraum aufräumen. Dafür soll er allen Weltraumschrott ,  und  aufheben.

Hinweis: Die letzte Zeile ist leer. Das Gitter ist immer 10 x 10 Felder groß. Der Roboter startet immer links oben.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



TEST 1

TEST 2

TEST 3

TEST 4

hebe Weltraumschrott auf

gehe vorwärts

drehe nach links

drehe nach rechts

drehe um

auf Weltraumschrott

falls
machewiederhole 10 mal:
mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

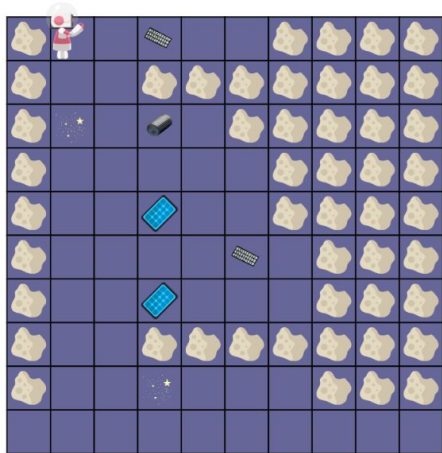
VERSION ☆☆☆



Programmiere den Roboter:

Der Roboter soll den Weltraum aufräumen. Dafür soll er allen Weltraumschrott ,  und  aufheben.

Hinweis: Die letzte Zeile ist leer. Der Roboter darf nicht gegen Asteroiden  fliegen. Der Schrott liegt in der Zeile immer vor einer Reihe Asteroiden. Das Feld hat 10 Zeilen und in der linken Spalte sind durchgängig Asteroiden. Der Roboter startet immer oben links.



TEST 1

TEST 2

TEST 3

TEST 4

hebe Weltraumschrott auf

gehe vorwärts

drehe nach links

drehe nach rechts

drehe um

auf Weltraumschrott

Asteroid voraus

falls
machewiederhole 10 mal:
machewiederhole bis
mache

Roboter-Programm

Noch 50 von 50 Bausteinen verfügbar.



Klasse 7-9



To exit full screen, press and hold

esc

Profil

Gruppenverwaltung

Lehrkräfte-Material

Wettbewerb: Jugendwettbewerb Informatik 2025 Runde 1: Jgst. 7

– 9

Dauer: 40 Minuten

Gruppenergebnisse

Ergebnisse meiner Gruppen für diesen Wettbewerb einsehen

Wusstest du schon, ...

... dass du Programme zwischen den verschiedenen Aufgabenversionen **kopieren** kannst? Nutze dazu einfach das Aufgabenmenü, oben rechts in der Aufgabe: , oder die Tastenkürzel Strg+C und Strg+V (Cmd+C und Cmd+V).

Teilnahme im Team

Logincode des Teampar

 **Jetzt im Team starten! (40 min)**

Wusstest du schon, ...

... dass du Programme zwischen den verschiedenen Aufgabenversionen **kopieren** kannst? Nutze dazu einfach das Aufgabenmenü, oben rechts in der Aufgabe: , oder die Tastenkürzel Strg+C und Strg+V (Cmd+C und Cmd+V).

Verbleibende Zeit: 0:39:55 (Lehrkräfte sind vom Zeitlimit ausgenommen.)

Route 66



Sternensammler



Finde die Rakete



Bunte Socken



Punktestand: 0 ★ / 16 ☆ (0 %)

Route 66

VERSION ☆☆☆

VERSION ☆☆☆☆

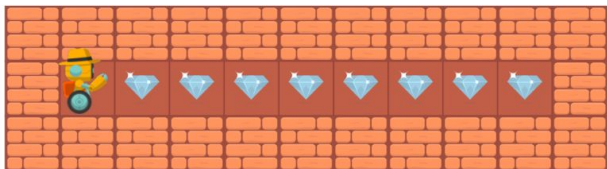
VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Diamanten  einsammeln.

Hinweis: Sobald er ein Feld mit einem Edelstein erreicht, wird dieser automatisch eingesammelt.



gehe vorwärts

Roboter-Programm

Noch 5 von 5 Bausteinen verfügbar.



vor Hindernis

wiederhole **bis**
mache

VERSION ☆☆☆

VERSION ☆☆☆

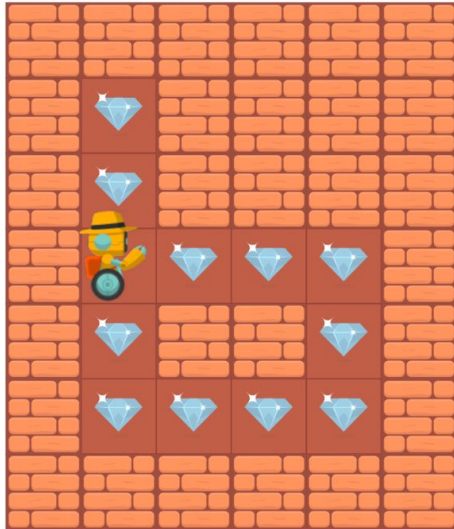
VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Diamanten  einsammeln.

Hinweis: Sobald er ein Feld mit einem Edelstein erreicht, wird dieser automatisch eingesammelt.



gehe vorwärts

drehe nach rechts

vor Hindernis

wiederhole bis
machewiederhole 10 mal:
mache

Roboter-Programm

Noch 8 von 8 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

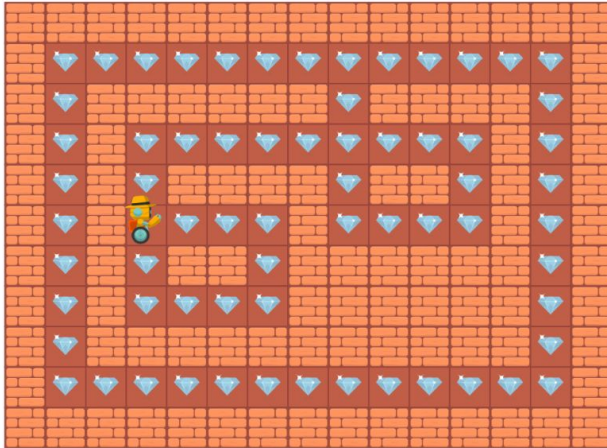
VERSION ☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Diamanten  einsammeln.

Hinweis: Sobald er ein Feld mit einem Edelstein erreicht, wird dieser automatisch eingesammelt.



gehe vorwärts

drehe nach rechts

vor Hindernis

wiederhole **bis**
machewiederhole **10** mal:
mache

Roboter-Programm

Noch 10 von 10 Bausteinen verfügbar.



Sternensammler

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆



Programmiere den Roboter:

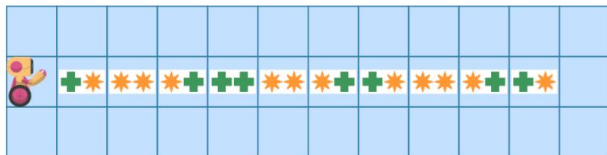
Der Roboter soll alle Dominosteine einsammeln, die einen oder zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".



WEITERE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen



gehe vorwärts

hebe Dominostein auf

auf gelbem Stern

auf grünem Kreuz

wiederhole 10 mal:

mache

falls

mache

Roboter-Programm

Noch 10 von 10 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die einen oder zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

VERSTECKTE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen

Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die einen oder zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

Weitere Hinweise:

Es müssen alle Dominosteine aufgesammelt werden, die einem der folgenden Muster entsprechen: , ,  oder .

Die Sensoren  auf gelbem Stern und  auf grünem Kreuz geben nur **True** oder **False** zurück und keine Anzahl.

TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



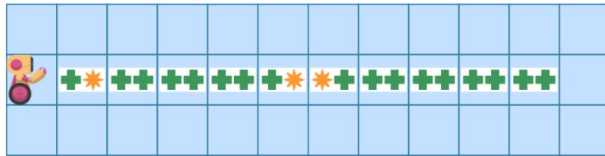
Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die genau einen Stern  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

[+ WEITERE HINWEISE](#)

Beachte: Dein Programm muss mit allen Testfällen



gehe vorwärts

hebe Dominostein auf

auf gelbem Stern

auf grünem Kreuz

wiederhole 10 mal:
machefalls
mache

und

nicht

Roboter-Programm

Noch 15 von 15 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3

TEST 4

TEST 5

TEST 6

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

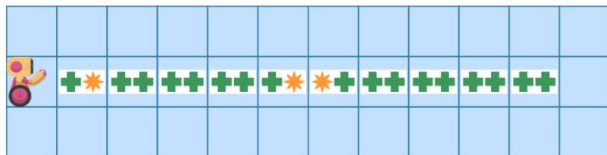
Der Roboter soll alle Dominosteine einsammeln, die genau einen Stern  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".



VERSTECKTE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen



Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die genau einen Stern  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

Weitere Hinweise:

Es müssen alle Dominosteine aufgesammelt werden, die einem der folgenden Muster entsprechen:  oder .

Die Sensoren  auf gelbem Stern und  auf grünem Kreuz geben nur oder zurück und keine Anzahl.

TEST 1

TEST 2

TEST 3

TEST 4

TEST 5

TEST 6



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die genau zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind ,  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

[+ WEITERE HINWEISE](#)

Beachte: Dein Programm muss mit allen Testfällen



gehe vorwärts

Roboter-Programm

Noch 15 von 15 Bausteinen verfügbar.

hebe Dominostein auf

auf gelbem Stern

auf grünem Kreuz

auf blauem Quadrat

wiederhole 10 mal:
machefalls
mache

und

nicht

TEST 1

TEST 2

TEST 3

TEST 4

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die genau zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind ,  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

VERSTECKTE HINWEISE

Beachte: Dein Programm muss mit allen Testfällen

Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll alle Dominosteine einsammeln, die genau zwei Sterne  haben.

Hinweis: Auf jedem Feld liegt genau ein Dominostein. Jeder Dominostein hat genau zwei Symbole. Mögliche Symbole sind ,  und . Vor dem Roboter liegen immer 10 Steine. Siehe auch "weitere Hinweise".

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

Weitere Hinweise:

Es müssen alle Dominosteine aufgesammelt werden, die dem folgenden Muster entsprechen:



Die Sensoren  auf gelbem Stern,  auf blauem Quadrat und  auf grünem Kreuz geben nur **True** oder **False** zurück und keine Anzahl.

TEST 1

TEST 2

TEST 3

TEST 4



Finde die Rakete

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Die Rakete ist in **Spalte 5** geparkt.



gehe vorwärts

Roboter-Programm

Noch 15 von 15 Bausteinen verfügbar.



Spalte des Roboters

wiederhole bis
mache

0

Spalte

setze Spalte auf



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist.

Hinweis: Zwischen Zahl und Zielspalte ist nie ein Asteroid

[+ WEITERE HINWEISE](#)

	1	2	3	4	5	6	7	8	9	10
1		7								

gehe vorwärts

gehe rückwärts

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze Spalte auf

Roboter-Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

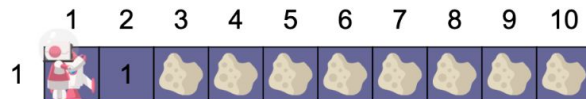
VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist.

Hinweis: Zwischen Zahl und Zielspalte ist nie ein Asteroid

[+ WEITERE HINWEISE](#)

gehe vorwärts

gehe rückwärts

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze Spalte auf

Roboter-Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist.

Hinweis: Zwischen Zahl und Zielspalte ist nie ein Asteroid

[+ WEITERE HINWEISE](#)

gehe vorwärts

gehe rückwärts

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze **Spalte** auf

Roboter-Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆

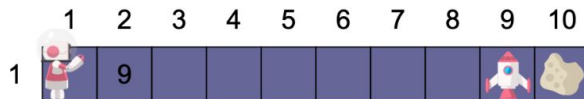


Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.

Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist.

Hinweis: Zwischen Zahl und Zielspalte ist nie ein Asteroid

[+ WEITERE HINWEISE](#)

gehe vorwärts

gehe rückwärts

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze **Spalte** auf

Roboter-Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

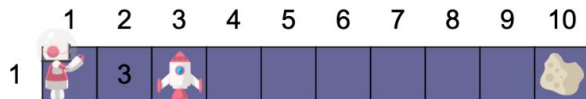
VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist.

Hinweis: Zwischen Zahl und Zielspalte ist nie ein Asteroid

[+ WEITERE HINWEISE](#)

gehe vorwärts

gehe rückwärts

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze Spalte auf

Roboter-Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4

TEST 5



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht

die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

[+ WEITERE HINWEISE](#)

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte eine ein

	1	2	3	4	5	6	7	8	9	10
1										
2		1								
3										
4										
5										
6										
7										
8										

TEST 1

TEST 2

TEST 3

TEST 4

gehe vorwärts

drehe nach rechts

drehe nach links

gehe rückwärts

vor Asteroiden

lies Zahl

Spalte des Roboters

wiederhole **bis**
mache

Spalte

setze Spalte auf

Roboter-Programm

Noch 35 von 35 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht

die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

VERSTECKTE HINWEISE

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein

	1	2	3	4	5	6	7	8	9	10
1										
2		1								
3										
4										
5										
6										
7										
8										

TEST 1

TEST 2

TEST 3

TEST 4



Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein Asteroid.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht

die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

VERSTECKTE HINWEISE

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein

	1	2	3	4	5	6	7	8	9	10
1										
2		4								
3										
4										
5										
6										
7										
8										



Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein Asteroid.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

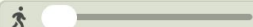
Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht

die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

VERSTECKTE HINWEISE

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein

	1	2	3	4	5	6	7	8	9	10
1										
2		1								
3										
4										
5										
6										
7										
8										

TEST 1

TEST 2

TEST 3

TEST 4



Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist

ist. In dieser Spalte steht die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein Asteroid.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt ist. In dieser Spalte steht

die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

VERSTECKTE HINWEISE

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein

	1	2	3	4	5	6	7	8	9	10
1										
2		8								
3										
4										
5										
6										
7										
8										

TEST 1

TEST 2

TEST 3

TEST 4



Aufgabenhinweise



Programmiere den Roboter:

Der Roboter soll zur Rakete  zurückkehren.
Auf dem Feld vor dem Roboter steht eine Zahl. Diese gibt die Spalte an, in der die Rakete geparkt

ist. In dieser Spalte steht die Rakete direkt vor dem ersten Asteroiden , den der Roboter beim nach unten laufen erreicht.

Hinweis: In Zeile 2 ist zwischen Zahl und Zielspalte nie ein Asteroid.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.



Bunte Socken

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe Sockenpaare** in der Schublade liegen.



In der Eingabe steht eine Übersicht der Sockenschublade:

- 1. Zeile Anzahl linke Socken in blau.
- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.

[+ WEITERE HINWEISE](#)

Eingabe:

54
62
48
53
64
67

Ausgabe:

schreibe

lies Zeile als ganze Zahl

falls
mache
sonst

=

wiederhole 10 mal:
mache

anzahlRechte

anzahlLinke

anzahlGelbePaare

setze anzahlRechte auf

setze anzahlLinke auf

setze anzahlGelbePaare auf

Programm

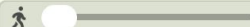
Noch 25 von 25 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. Zeile** Anzahl linke Socken in blau.
- **2. Zeile** Anzahl rechte Socken in blau.
- **3. Zeile** Anzahl linke Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

54
62
48
53
64
67

Ausgabe:



Aufgabenhinweise



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. Zeile** Anzahl linke Socken in blau.
- **2. Zeile** Anzahl rechte Socken in blau.
- **3. Zeile** Anzahl linke Socken in gelb.
- **4. Zeile** Anzahl rechte Socken in gelb.
- **5. Zeile** Anzahl linke Socken in rot.
- **6. Zeile** Anzahl rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von

Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- 1. Zeile Anzahl linke Socken in blau.
- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

54
62
48
53
64
67

Ausgabe:



Aufgabenhinweise



- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.
- 4. Zeile Anzahl rechte Socken in gelb.
- 5. Zeile Anzahl linke Socken in rot.
- 6. Zeile Anzahl rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- 1. Zeile Anzahl linke Socken in blau.
- 2. Zeile Anzahl rechte
- 3. Zeile Anzahl linke Socken in gelb

VERSTECKTE HINWEISE

Eingabe:

23
26
37
10
3
5

Ausgabe:



Aufgabenhinweise



- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.
- 4. Zeile Anzahl rechte Socken in gelb.
- 5. Zeile Anzahl linke Socken in rot.
- 6. Zeile Anzahl rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst** zur Verfügung.

*Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.*

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- 1. Zeile Anzahl linke Socken in blau.
- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

50
0
50
0
0
50

Ausgabe:



Aufgabenhinweise



- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.
- 4. Zeile Anzahl rechte Socken in gelb.
- 5. Zeile Anzahl linke Socken in rot.
- 6. Zeile Anzahl rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gelbe****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- 1. Zeile Anzahl linke Socken in blau.
- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

72
93
14
29
31
44

Ausgabe:



Aufgabenhinweise



- 2. Zeile Anzahl rechte Socken in blau.
- 3. Zeile Anzahl linke Socken in gelb.
- 4. Zeile Anzahl rechte Socken in gelb.
- 5. Zeile Anzahl linke Socken in rot.
- 6. Zeile Anzahl rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst**-Bausteine zur Verfügung.

*Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.*

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige Sockenpaare** in der Schublade liegen.



In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken

[+ WEITERE HINWEISE](#)

Eingabe:

54
62
48
53
64
67

Ausgabe:

schreibe

lies Zeile als ganze Zahl

Ende der Eingabe

falls

mache

sonst

==

wiederhole 10 mal:

mache

wiederhole bis

mache

0

1 + 1

anzahlRechte

anzahlLinke

anzahlPaare

Programm

Noch 30 von 30 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3

TEST 4

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige Sockenpaare** in der Schublade liegen.



In der Eingabe steht eine Übersicht der Sockenschublade:

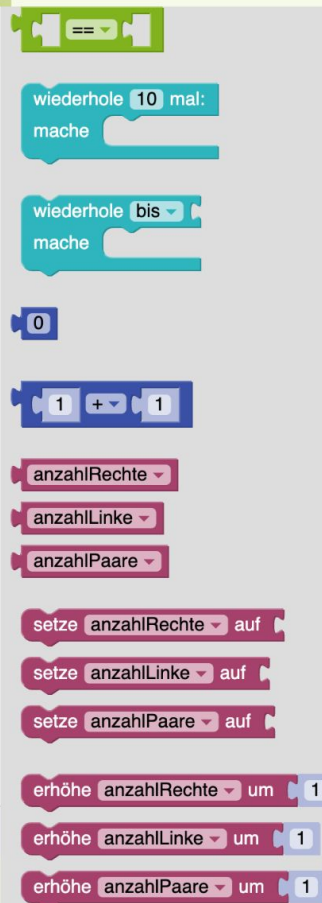
- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken

[+ WEITERE HINWEISE](#)

Eingabe:

54
62
48
53
64
67

Ausgabe:



Programm

Noch 30 von 30 Bausteinen verfügbar.



TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

54
62
48
53
64
67

Ausgabe:



Aufgabenhinweise



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.
- **5. und 6. Zeile** Anzahl linke und rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von

Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache**-Bausteine zur Verfügung.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige****Sockenpaare** in der Schublade liegen.In der Eingabe steht eine Übersicht der
Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.

[VERSTECKE HINWEISE](#)

Eingabe:

54
62
48
53
64
67

Ausgabe:



Aufgabenhinweise



- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.
- **5. und 6. Zeile** Anzahl linke und rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls**, **mache**, **sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

23
26
37
10
3
5

Ausgabe:



Aufgabenhinweise



- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.
- **5. und 6. Zeile** Anzahl linke und rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls**, **mache**, **sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

50
0
50
0
0
50

Ausgabe:



Aufgabenhinweise



- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.
- **5. und 6. Zeile** Anzahl linke und rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



Schreibe ein Programm:

Das Programm soll ausgeben, wie viele **gleichfarbige****Sockenpaare** in der Schublade liegen.

In der Eingabe steht eine Übersicht der Sockenschublade:

- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.

VERSTECKTE HINWEISE

Eingabe:

72
93
14
29
31
44

Ausgabe:



Aufgabenhinweise



- **1. und 2. Zeile** Anzahl linke und rechte Socken in blau.
- **3. und 4. Zeile** Anzahl linke und rechte Socken in gelb.
- **5. und 6. Zeile** Anzahl linke und rechte Socken in rot.

Hinweis: Ein Sockenpaar besteht aus einem linken und einem rechten Socken derselben Farbe.

Variablen:

Bevor du eine Variable benutzen kannst, musst du ihr einen Wert zuweisen. Zum Beispiel:

setze Variablen Name auf 0

Eingeschränkte Bausteine:

Um eine reine Sequenz an Anweisungen oder eine Lösung, die nur auf das Bestehen von Testfällen abzielt, zu vermeiden, stehen maximal drei **schreibe**-Bausteine und maximal drei **falls mache sonst** zur Verfügung.

Beachte: Dein Programm muss mit allen Testfällen zurechtkommen.

TEST 1

TEST 2

TEST 3

TEST 4



Klasse 10+



Wettbewerb: Jugendwettbewerb Informatik 2025 Runde 1: ab Jgst. 10

Dauer: 60 Minuten

Gruppenergebnisse

Ergebnisse meiner Gruppen für diesen Wettbewerb einsehen

Wusstest du schon, ...

... dass du Programme zwischen den verschiedenen Aufgabenversionen **kopieren** kannst? Nutze dazu einfach das Aufgabenmenü, oben rechts in der Aufgabe: , oder die Tastenkürzel Strg+C und Strg+V (Cmd+C und Cmd+V).

Teilnahme im Team

Logincode des Teampar

 **Jetzt im Team starten! (60 min)**

Aufgabenversionen kopieren kannst. Nutze dazu einfach das Aufgabenmenü, oben rechts in der Aufgabe: , oder die Tastenkürzel Strg+C und Strg+V (Cmd+C und Cmd+V).

Verbleibende Zeit: 0:59:53 (Lehrkräfte sind vom Zeitlimit ausgenommen.)

Weltraumaufräumer



Chatbot



Sternensammler



Robo.Go



Holz gerecht verteilen



Bunte Socken



Punktestand: 0 ★ / 24 ☆ (0 %)

Robo.Go

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



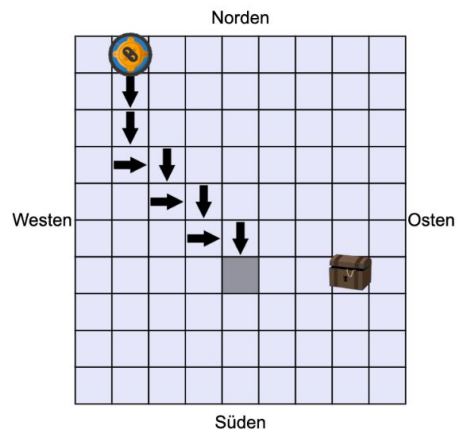
Programmiere den Roboter:



bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Osten**, wie der Roboter bereits vorher nach Osten **gegangen** ist.

[+ WEITERE HINWEISE](#)

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.

gehe nach rechts

gehe nach unten

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

wiederhole solange

mache

wiederhole 10 mal:

mache

0

falls

mache

AnzahlSchritteRechts

TEST 1

TEST 2

TEST 3

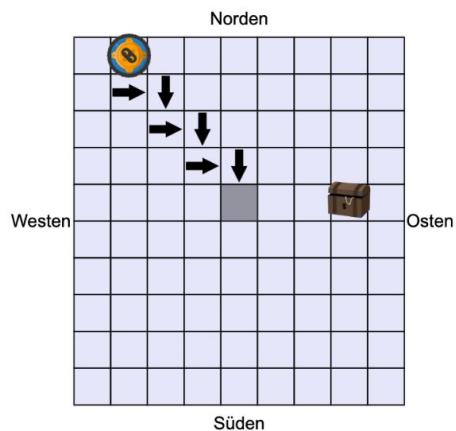




Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Osten**, wie der Roboter bereits vorher nach Osten **gegangen** ist.

 [WEITERE HINW](#)

WEITERE HINWEISE



TEST 3



wiederhole solange
mache

wiederhole 10 mal:
mache

0

 falls
mache

AnzahlSchritteRechts ▼

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



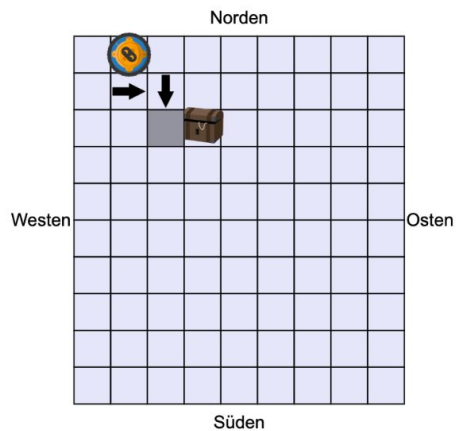
Programmiere den Roboter:



bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Osten**, wie der Roboter bereits vorher nach Osten **gegangen** ist.

[+ WEITERE HINWEISE](#)

TEST 1

TEST 2

TEST 3



gehe nach rechts

gehe nach unten

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

wiederhole solange

mache

wiederhole 10 mal:

mache

0

falls

mache

AnzahlSchritteRechts

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



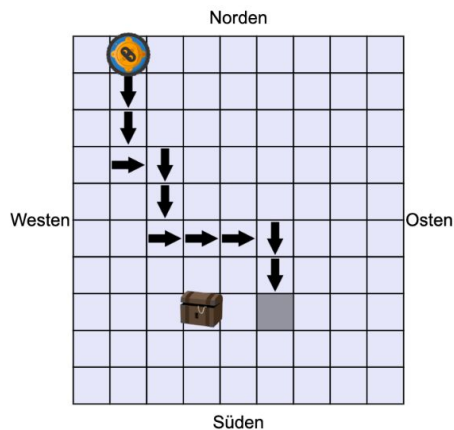
Programmiere den Roboter:



bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Westen**, wie der Roboter vorher Richtung **Osten abgebogen** ist (also von einem Pfeil nach  **WEITERE HINWEISE** nach rechts gewechselt ist)



gehe nach rechts

gehe nach unten

gehe nach links

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

wiederhole solange
machewiederhole 10 mal:
mache

0

falls
mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3

VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



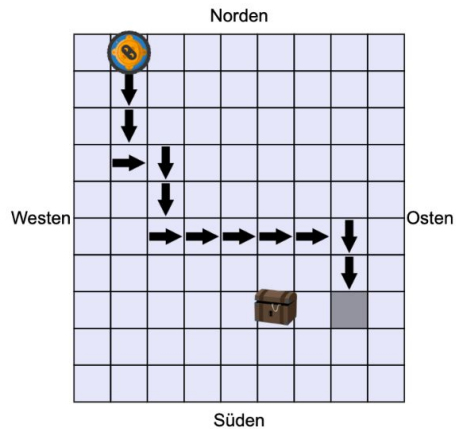
Programmiere den Roboter:



bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Westen**, wie der Roboter vorher Richtung **Osten** **abgebogen** ist (also von einem Pfeil nach  **WEITERE HINWEISE** nach rechts gewechselt ist)



gehe nach rechts

gehe nach unten

gehe nach links

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

wiederhole solange
machewiederhole 10 mal:
mache

0

falls
mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



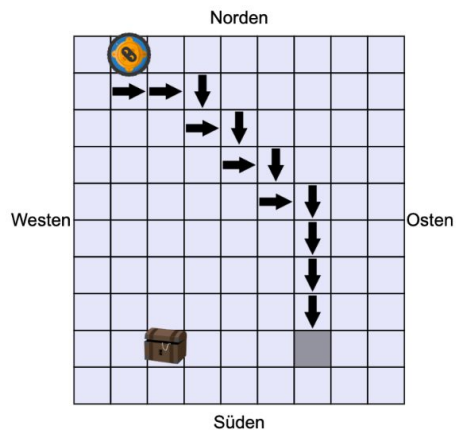
Programmiere den Roboter:



bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position  so viele Schritte nach **Westen**, wie der Roboter vorher Richtung **Osten** **abgebogen** ist (also von einem Pfeil nach  **WEITERE HINWEISE** nach rechts gewechselt ist)



gehe nach rechts

gehe nach unten

gehe nach links

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

wiederhole **solange** machewiederhole **10** mal: mache

0

falls mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.

TEST 1

TEST 2

TEST 3



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆

📄 Programmiere den Roboter:

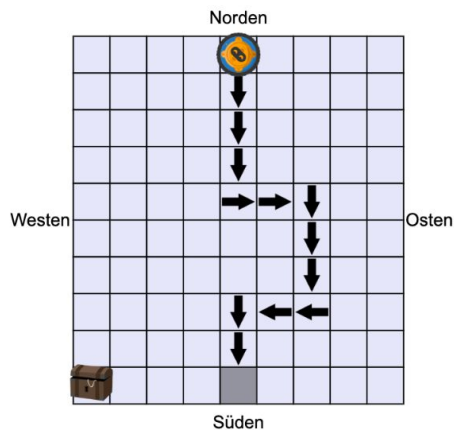


bergen. Dafür

muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position vier Schritte nach Westen, wenn der Roboter mindestens so oft nach Westen **abgebogen** ist, wie nach Osten. [+ WEITERE HINWEISE](#)

Truhe vier Schritte nach Osten



TEST 1

TEST 2

TEST 3

TEST 4

gehe nach rechts

gehe nach unten

gehe nach links

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

auf Pfeil nach links

wiederhole solange

mache

wiederhole 10 mal:

mache

falls

mache

sonst

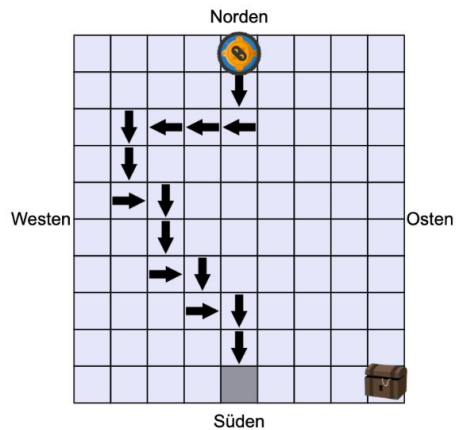
Roboter-Programm

Noch 40 von 40 Bausteinen verfügbar.



Die Schatztruhe liegt von der letzten Position vier Schritte nach Westen, wenn der Roboter mindestens so oft nach Westen **abgebogen** ist, wie nach Osten. [WEITERE HINWEISE](#)

WEITERE HINWEISE



TEST 4



- falls
- mache
- sonst

Noch 40 von 40 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆

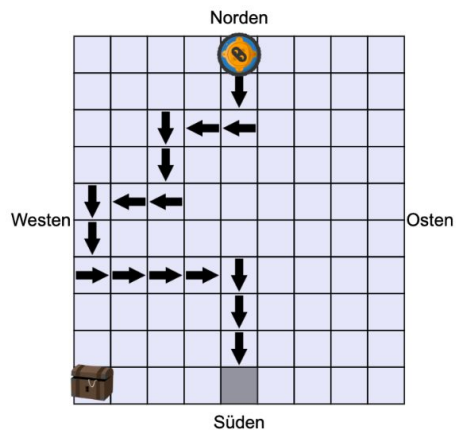
📄 Programmiere den Roboter:



Der Roboter soll die Schatztruhe bergen. Dafür muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position vier Schritte nach Westen, wenn der Roboter mindestens so oft nach Westen **abgebogen** ist, wie nach Osten. [+ WEITERE HINWEISE](#)

Truhe vier Schritte nach Osten



TEST 1

TEST 2

TEST 3

TEST 4



gehe nach rechts

gehe nach unten

gehe nach links

auf Pfeil

auf Pfeil nach unten

auf Pfeil nach rechts

auf Pfeil nach links

wiederhole solange

mache

wiederhole 10 mal:

mache

falls

mache

sonst

Roboter-Programm

Noch 40 von 40 Bausteinen verfügbar.





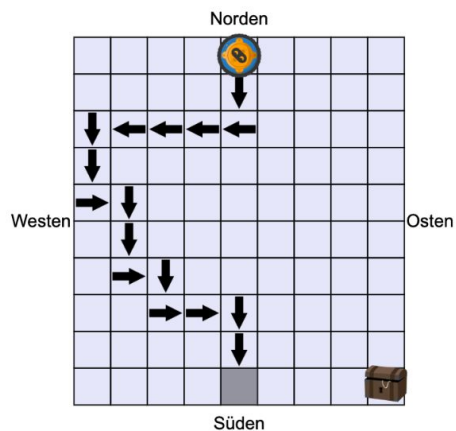
 Programmiere den Roboter:



Der Roboter soll die Schatztruhe  bergen. Dafür muss er dem Pfad folgen, den ihm die Pfeile vorgeben. Der Pfad kodiert also, wo die Truhe zu finden ist.

Die Schatztruhe liegt von der letzten Position vier Schritte nach Westen, wenn der Roboter mindestens so oft nach Westen **abgebogen** ist, wie nach Osten. [WEITERE HINWEISE](#)

WEITERE HINWEISE



TEST 1

TEST 2

TEST 3

TEST 4



gehe nach rechts

gehe nach unten

gehe nach links

☛ auf Pfeil

☛ auf Pfeil nach unten

☛ auf Pfeil nach rechts

☛ auf Pfeil nach links

wiederhole solange
mache

wiederhole 10 mal:
mache

- falls
- mache
- sonst

Roboter-Programm

Noch 40 von 40 Bausteinen verfügbar.



Holz gerecht verteilen

VERSION ☆☆☆

VERSION ☆☆☆

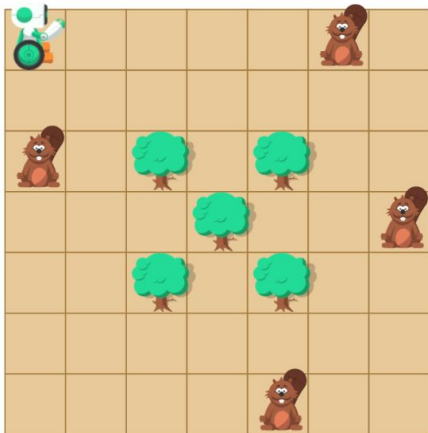
VERSION ☆☆☆

 Programmiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und jedem Biber gerecht einen Holzstapel übergeben.

Hinweis: Die Biber stehen immer am Rand des Gitters. Das Gitter ist immer 7x7 Felder groß. Der Roboter startet in der



TEST 1

TEST 2

gehe vorwärts

drehe nach links

drehe nach rechts

samme Holzstapel ein

Holzstapel übergeben

Anzahl Holzstapel

bei Biber

wiederhole mal:

mache

falls

mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆

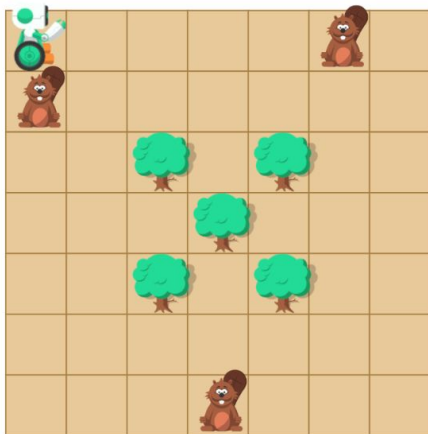


Programmiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und jedem Biber gerecht einen Holzstapel übergeben.

Hinweis: Die Biber stehen immer am Rand des Gitters. Das Gitter ist immer 7x7 Felder groß. Der Roboter startet in der



TEST 1

TEST 2



gehe vorwärts

drehe nach links

drehe nach rechts

sammler 0 Holzstapel ein

Holzstapel übergeben

Anzahl Holzstapel

bei Biber

wiederhole 10 mal:

mache

falls

mache

0

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



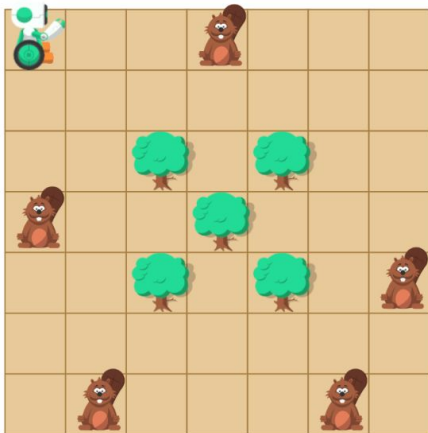
Programmiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und gerecht auf die fünf Biber verteilen. Jeder Biber bekommt mindestens einen Holzstapel und alle Biber bekommen gleich viel.

[+ WEITERE HINWEISE](#)

Es liegen immer so viele Holzstapel an der Startposition, dass



TEST 1

TEST 2

gehe vorwärts

drehe nach links

drehe nach rechts

samme Holzstapel ein

Holzstapel übergeben

übergebe Holzstapel

Anzahl Holzstapel

bei Biber

Anzahl Holzstapel in der Tasche

wiederhole mal:

mache

wiederhole bis

mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆



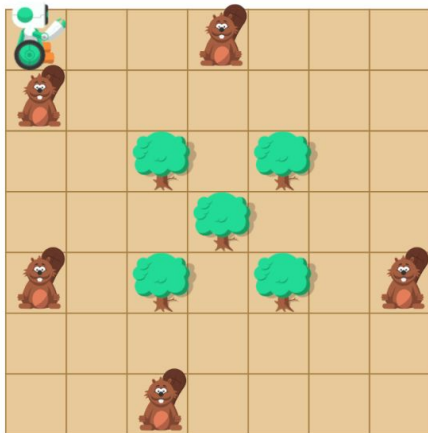
📄 Programmeiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und gerecht auf die fünf Biber verteilen. Jeder Biber bekommt mindestens einen Holzstapel und alle Biber bekommen gleich viel.

[+ WEITERE HINWEISE](#)

Es liegen immer so viele Holzstapel an der Startposition, dass



TEST 1

TEST 2



gehe vorwärts

drehe nach links

drehe nach rechts

samme Holzstapel ein

Holzstapel übergeben

übergebe Holzstapel

Anzahl Holzstapel

bei Biber

Anzahl Holzstapel in der Tasche

wiederhole mal:

mache

wiederhole bis

mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆☆

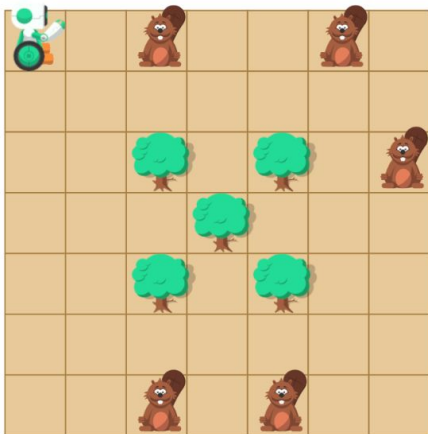


Programmiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und gerecht auf die fünf Biber verteilen. Jeder Biber bekommt mindestens einen Holzstapel und alle Biber bekommen gleich viel.

Es liegen immer so viele Holzstapel an der Startposition, dass



TEST 1

TEST 2

gehe vorwärts

drehe nach links

drehe nach rechts

samme  0 Holzstapel ein

Holzstapel übergeben

Anzahl Holzstapel

bei Biber

Anzahl Holzstapel in der Tasche

falls
machewiederhole  10 mal:

mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



VERSION ☆☆☆

VERSION ☆☆☆

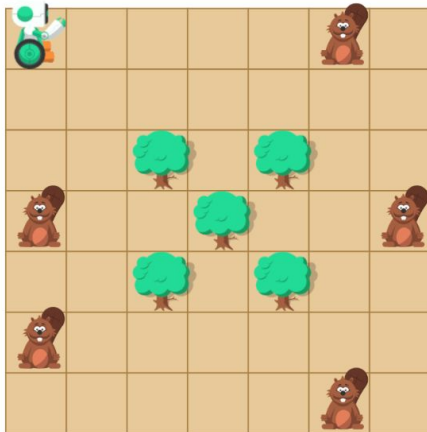
VERSION ☆☆☆

 Programmiere den Roboter:

Der Roboter soll die Biber  füttern.

Dafür muss er die Holzstapel  an der Startposition einsammeln und gerecht auf die fünf Biber verteilen. Jeder Biber bekommt mindestens einen Holzstapel und alle Biber bekommen gleich viel.

Es liegen immer so viele Holzstapel an der Startposition, dass



TEST 1

TEST 2



gehe vorwärts

drehe nach links

drehe nach rechts

samme  0 Holzstapel ein

Holzstapel übergeben

Anzahl Holzstapel

bei Biber

Anzahl Holzstapel in der Tasche

falls
machewiederhole  10 mal:

mache

Roboter-Programm

Noch 20 von 20 Bausteinen verfügbar.



