

PROGRAMMIEREN LERNEN MIT DER STICKMASCHINE

Stand: 02.04.2025

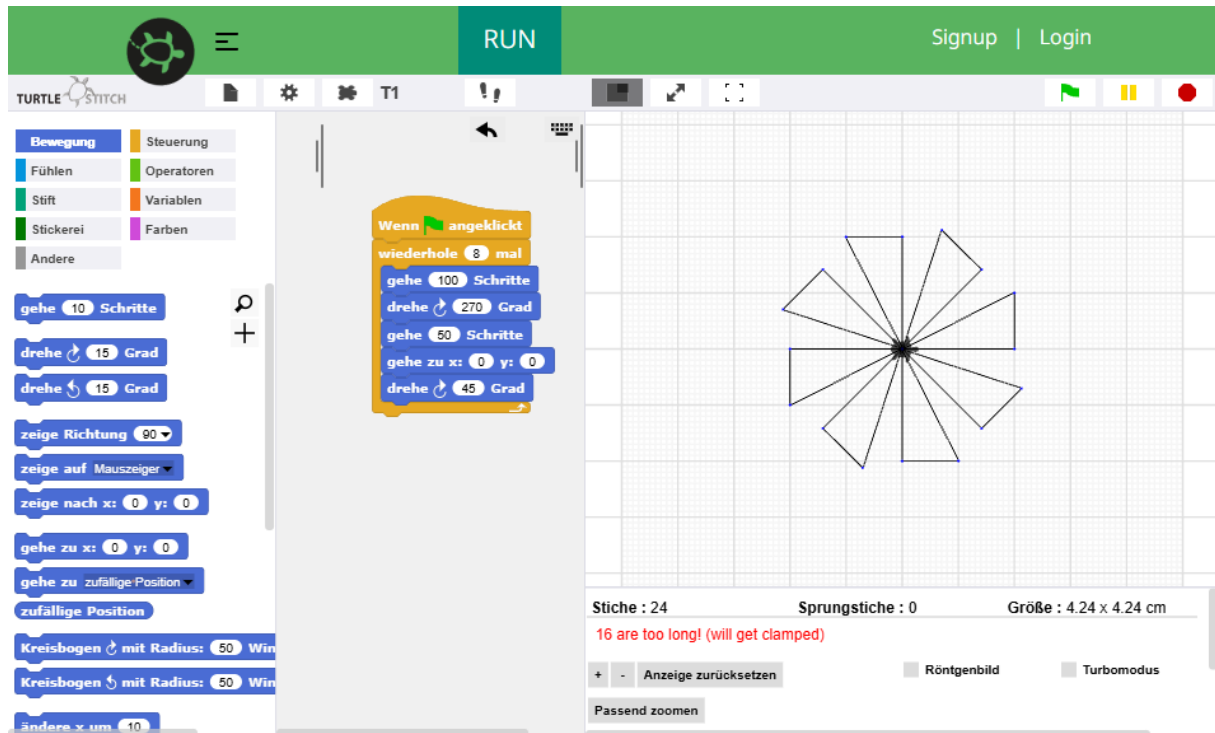


Vorwort zum Girls Day

Das Ziel des Workshops ist, einen Einkaufsbeutel selbst mit Stickereien zu verzieren. Der Workshop ist in 3 Teile geteilt. Im ersten Teil (2 Stunden, vormittags) lernen wir, wie man Muster durch Programmierung erstellt. Hierfür ist dieses Arbeitsheft, in dem primär vorhandene Muster nachprogrammiert werden. Im zweiten Teil (2 Stunden, nachmittags) erstellen wir selbst ein einfarbiges Muster. Im dritten Teil (1 Stunde, nachmittags) drucken wir das Muster aus Teil 2 und erweitern unser Muster mit Objekten aus mehreren Farben. Da das Stickern aller Muster etwas dauert und unsere Maschine nur eine Farbe auf einmal sticken kann, können wir das mehrfarbige Muster jedoch leider nicht einsticken.

Aufgabe 1: Die Programmierumgebung

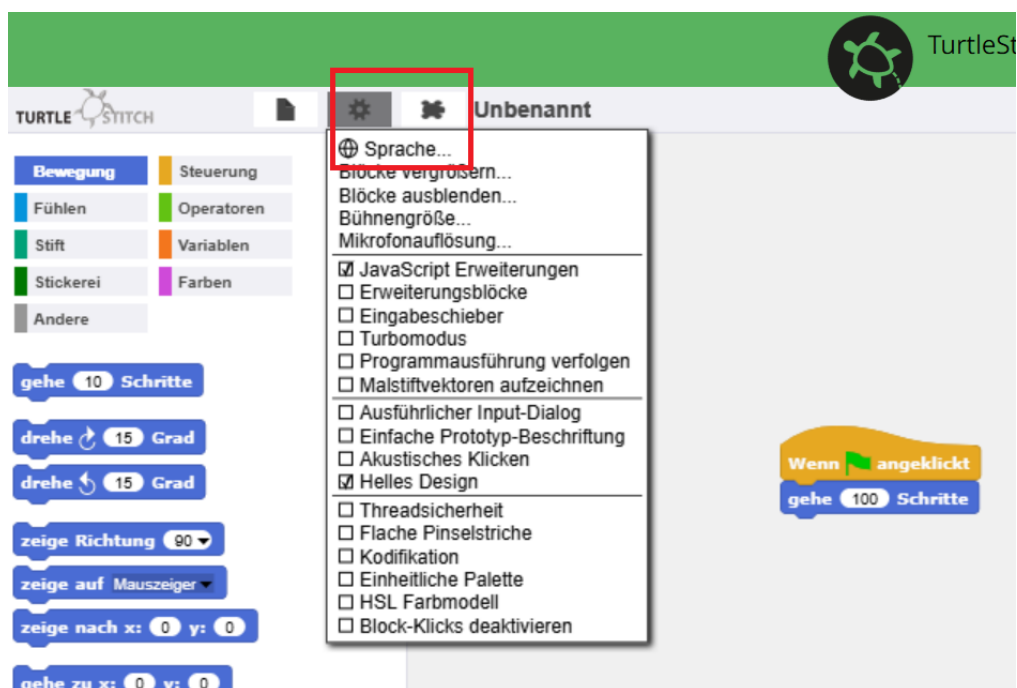
Wir werden in dieser Aufgabe mit der Programmierumgebung Turtle Stitch arbeiten. Diese ist speziell für die Entwicklung von Stickmustern programmiert worden.



Aufgabe 1a)

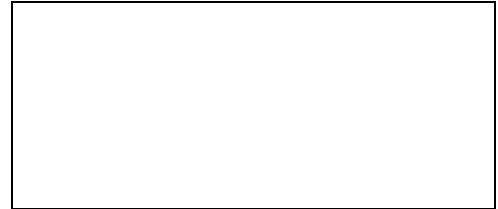
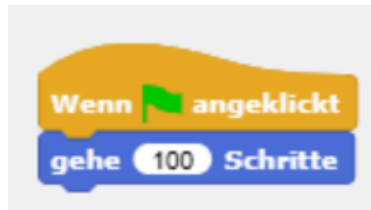
Öffne die Seite <https://www.turtlestitch.org/run> in Firefox.

Stelle ggf. die Sprache auf Deutsch um:



Aufgabe 1b)

Übernimm das Programm links in Turtlestitch. Der obere Block ist im orangenen Bereich („Steuerung“), der untere Block im blauen Bereich („Bewegung“). Drücke danach auf die grüne Fahne rechts oben über der Ausgabefläche und lasse so das Ergebnis in diese einzeichnen. Zeichne es im weißen Kasten rechts nach.



Aufgabe 1c)

Unter dem Ergebnis werden Informationen angezeigt. Wie viele Stiche und wie viel Platz werden für das Muster benötigt?



Aufgabe 1d)

Übernimm das Programm im Bild, führe es erneut aus und zeichne das Ergebnis rechts ein.



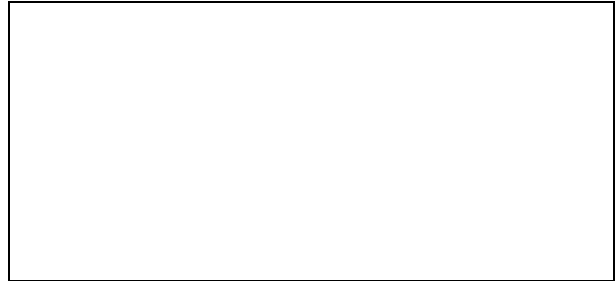
Aufgabe 1e)

Wie viele Stiche und wie viel Platz werden für das Muster benötigt?



Aufgabe 1f)

Ändere das Programm wie im Bild links gezeigt, führe es aus und zeichne das Ergebnis rechts ein.



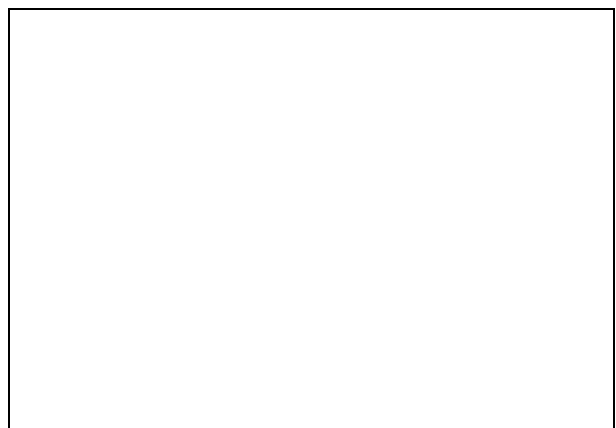
Aufgabe 1g)

Beschreibe, was die Funktionen „gehe zu“ und „löschen“ machen

An empty rectangular box for describing the functions 'gehe zu' and 'löschen'.

Aufgabe 1h)

Übernimm das Programm im Bild links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 1i)

Beschreibe, was die x und y Positionen bedeuten. Wie unterscheiden sich x und y?

An empty rectangular box for describing the x and y positions.

Aufgabe 1j)

Wie viele Stiche und wie viel Platz benötigt das Muster?

Aufgabe 1k)

Ändere die Zahl im grünen Kasten auf 5 wie im Bild gezeigt:



Führe das Programm erneut aus. Wie viele Stiche benötigt das Muster jetzt?

Aufgabe 1l)

Übernimm das Programm links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 1m)

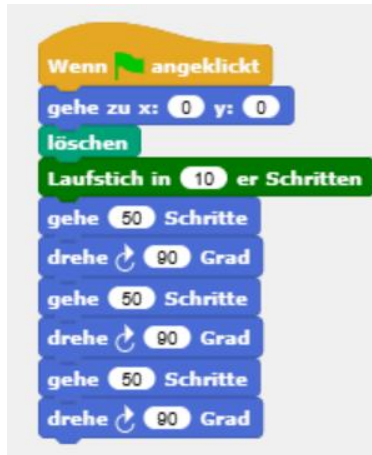
Ändere die beiden Zahlen, die man dem Kreuzstich übergeben kann.

Beschreibe für jede Zahl, wie sie das Muster verändert.

Aufgabe 2: Einfache Muster mit Wiederholungen

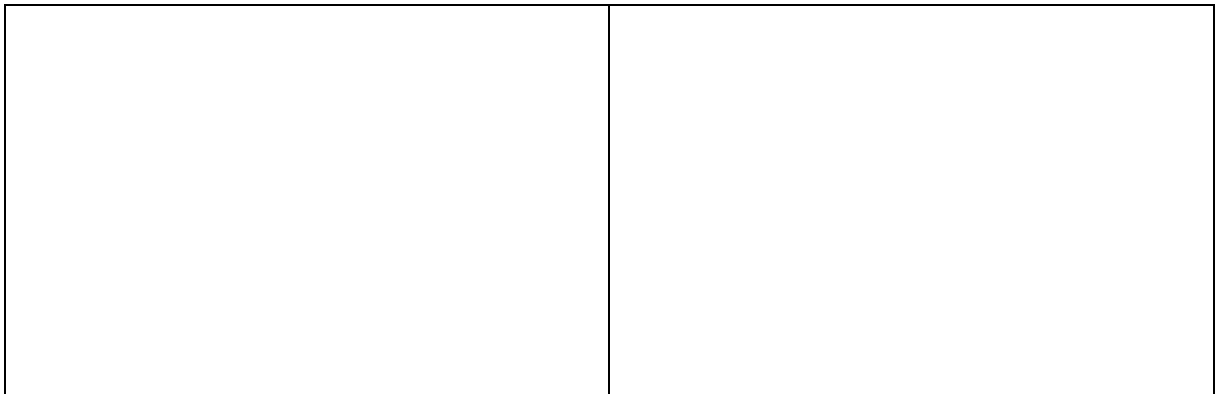
Aufgabe 2a)

Übernimm das Programm links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 2b)

Führe das Programm noch zweimal aus. Zeichne das Ergebnis nach jeder Ausführung ein.



Aufgabe 2c)

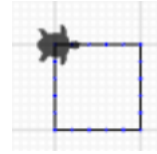
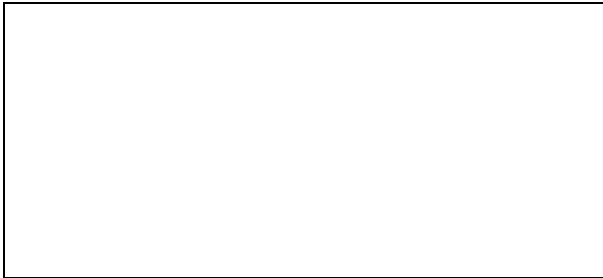
Beschreibe, wie du erreichen kannst, dass immer das gleiche Bild herauskommt.

Tipp (umgedreht): an ,bungewæd, ꝥrææd wɪ ælhæd ællæ .ɹɪ ænæps



Aufgabe 2d)

Programmiere das rechte Muster nach. Schreibe dein Programm links in den Kasten. Du darfst dabei Abkürzungen verwenden (z.B. „G100“ statt „gehe 100 Schritte“)



Aufgabe 2e)

Übernimm das Programm links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 2f)

Ändere die Anzahl der Wiederholungen auf 6:

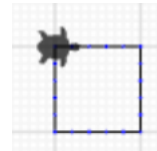
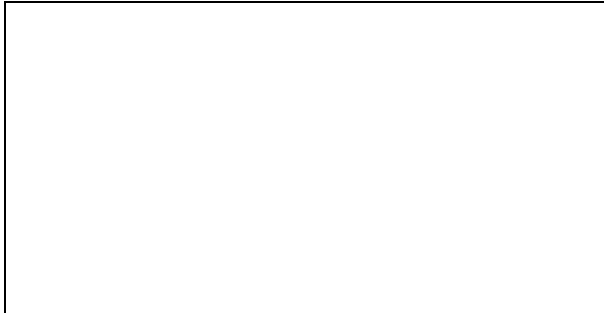


Beschreibe, was sich verändert hat



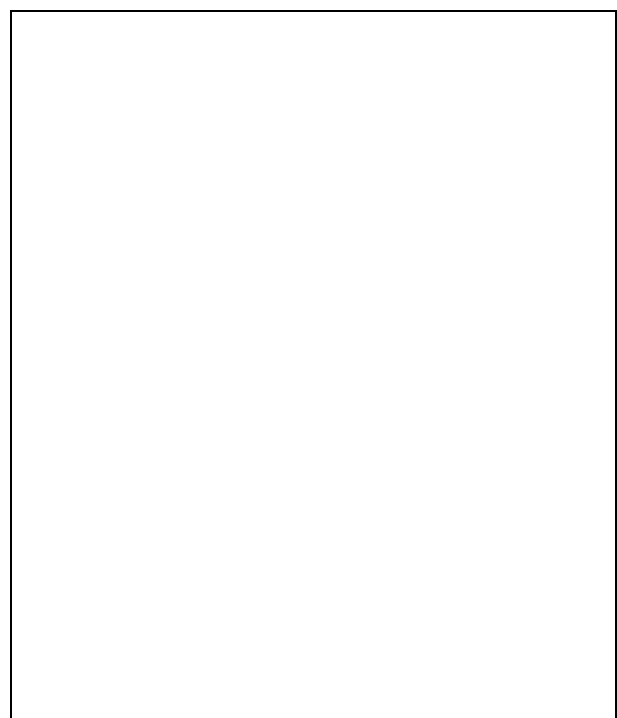
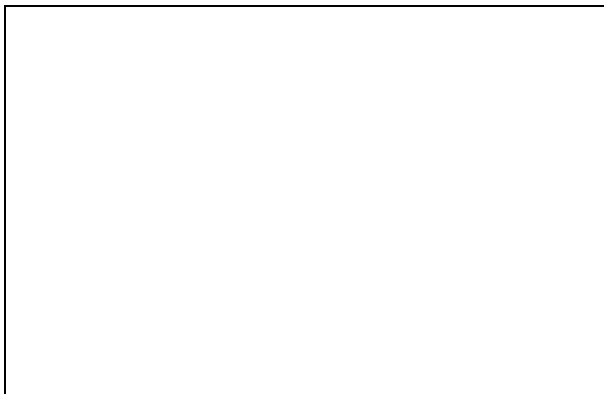
Aufgabe 2g)

Zeichne das Rechteck erneut, dieses Mal aber mit der Verwendung von Wiederholungen. Schreibe dein Programm in den Kasten links. Abkürzungen sind wieder erlaubt – dies gilt auch für alle folgenden Aufgaben.



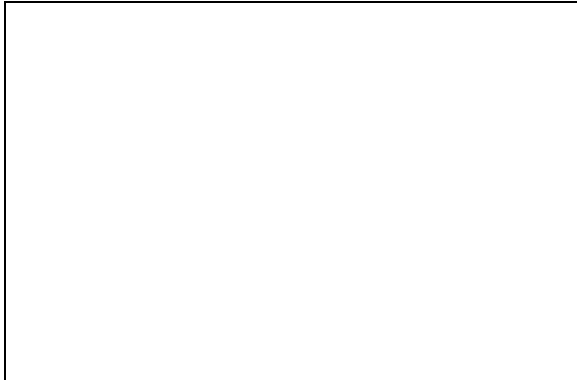
Aufgabe 2h)

Programmiere das Muster rechts nach. Schreibe dein Programm in den Kasten links.



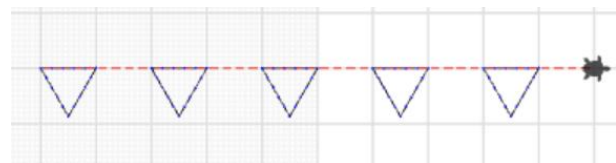
Aufgabe 2j)

Beschreibe im Kasten links, wie du das Programm aus der vorherigen Aufgabe anpassen musst, damit das Muster rechts entsteht.



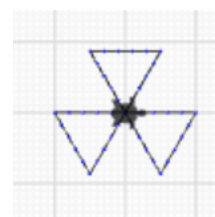
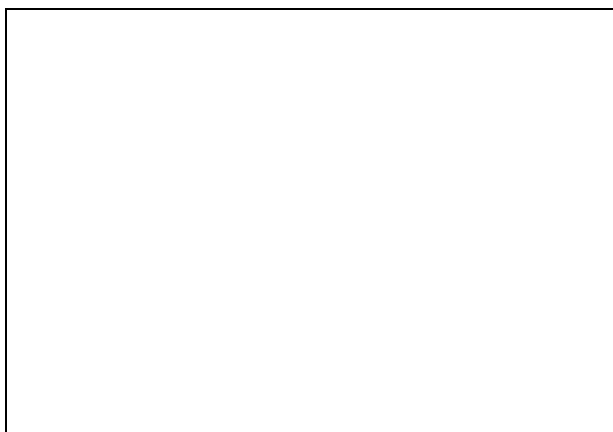
Aufgabe 2k)

Programmiere das Muster rechts nach. Schreibe dein Programm in den Kasten links.



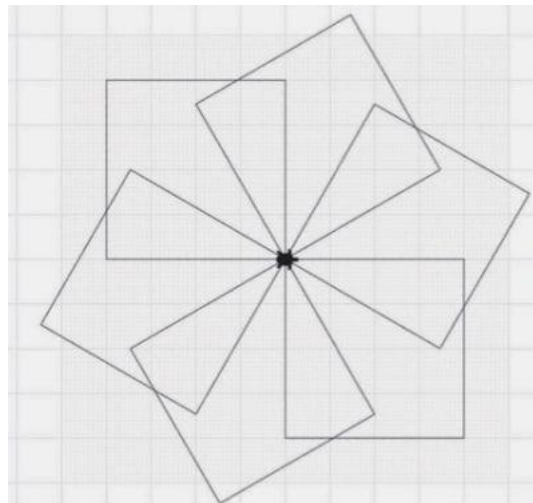
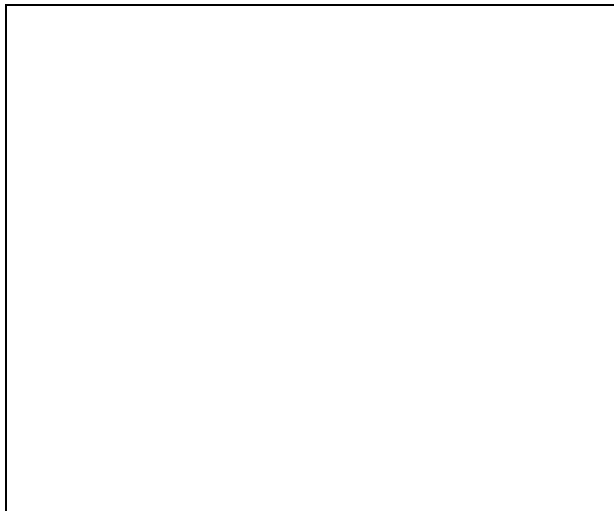
Aufgabe 2l)

Programmiere das Muster rechts nach. Schreibe dein Programm in den Kasten links.

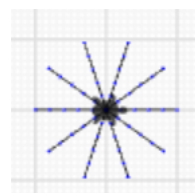
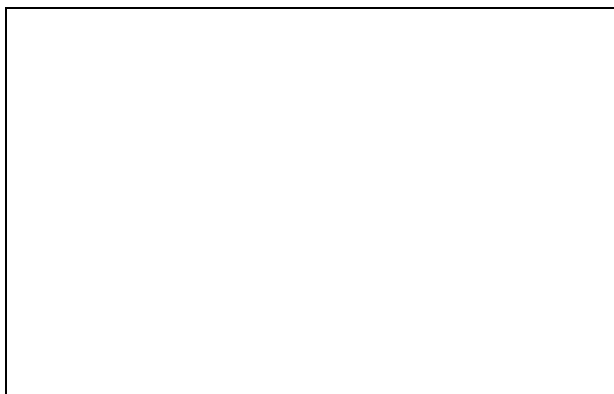


Aufgabe 2m)

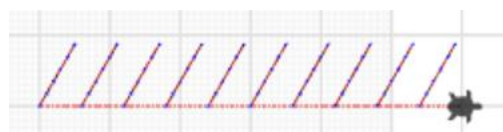
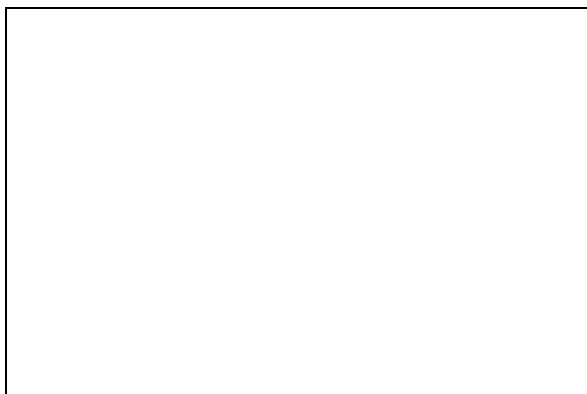
Programmiere das Muster rechts nach. Schreibe dein Programm in den Kasten links.

**Aufgabe 2n)**

Programmiere das Muster rechts mit 10 Linien nach. Schreibe dein Programm in den Kasten links.

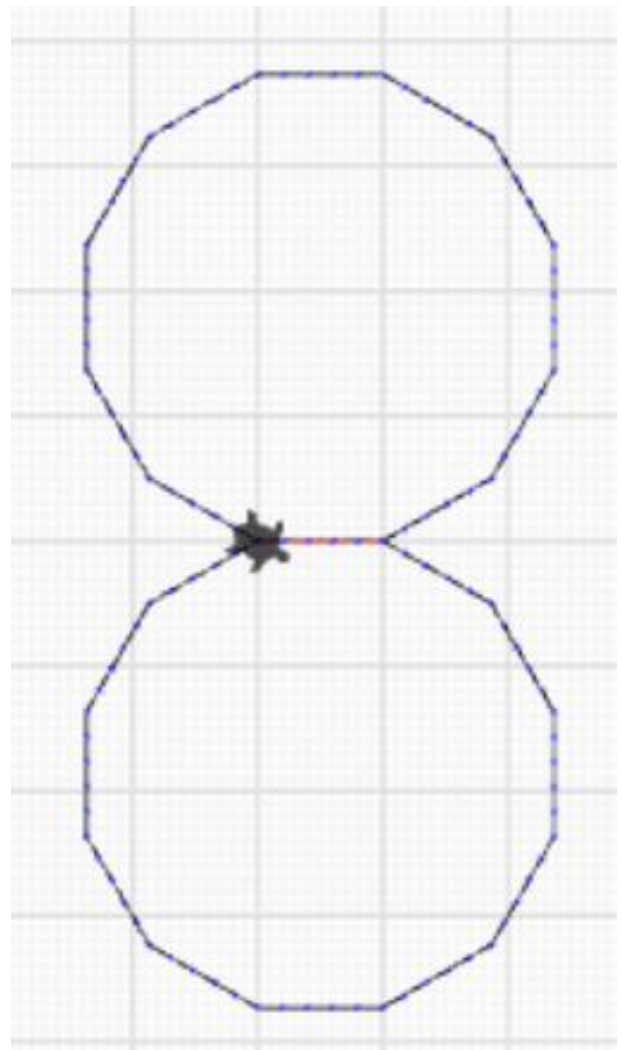
**Aufgabe 2o)**

Programmiere das Muster rechts mit 10 Linien nach. Schreibe dein Programm in den Kasten.



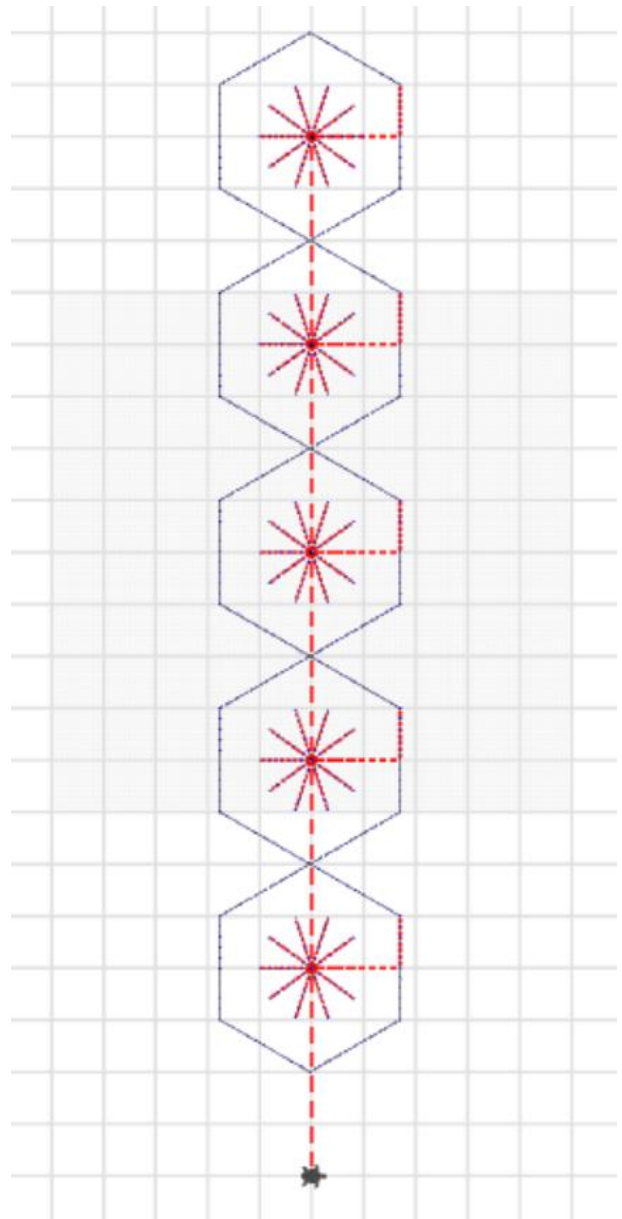
Aufgabe 2p)

Programmiere das Muster rechts mit 23 Linien nach. Achte darauf, dass keine Linie doppelt gestickt werden muss. Schreibe dein Programm in den Kasten links.



Aufgabe 2q)

Programmiere das Muster rechts nach. Schreibe dein Programm in den Kasten links.



Aufgabe 3: Eigene Blöcke ohne und mit Parameter

Die bisherigen Programme waren sehr lang und hatten viele Wiederholungen. Das macht sie schwer lesbar und anpassbar. Das wollen wir nun verbessern.

Aufgabe 3a)

Klicke im Bereich „Andere“ auf „Neuer Block“. Nenne ihn „Viereck“. Dann sollte ein Editorfenster erscheinen:



Aufgabe 3b)

Fülle den neuen Block mit dem links gezeigten Inhalt.

Du kannst ihn anschließend wie rechts gezeigt als einzelnen Block benutzen.

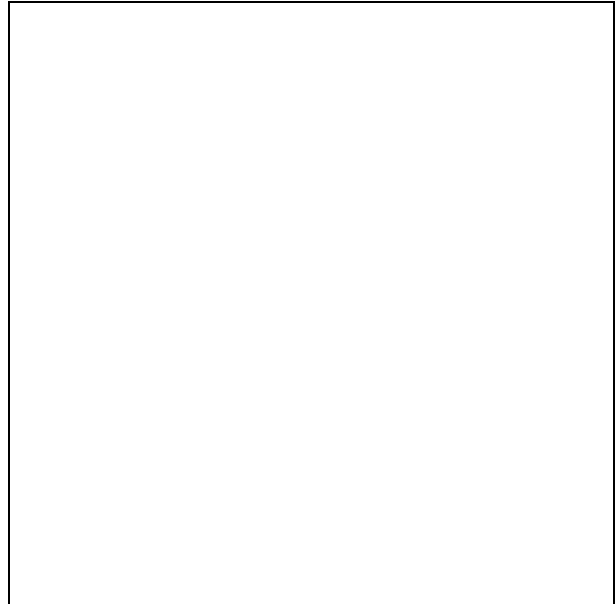
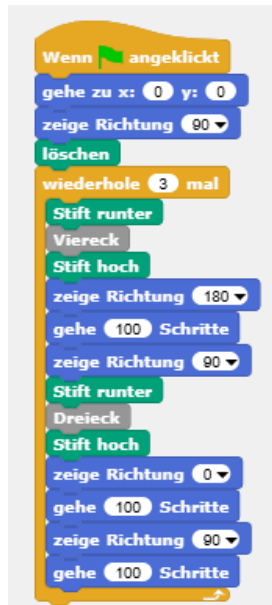


Baue den Programmtext mit dem eigenem Block nach und teste es.

Aufgabe 3c)

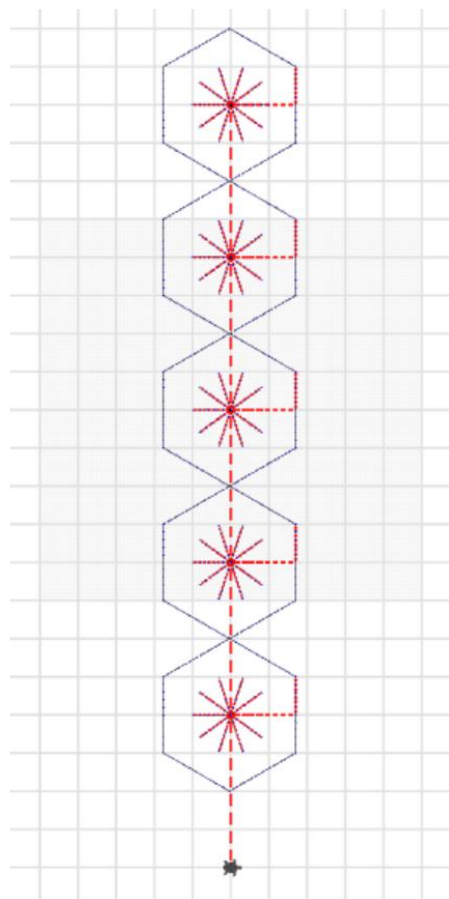
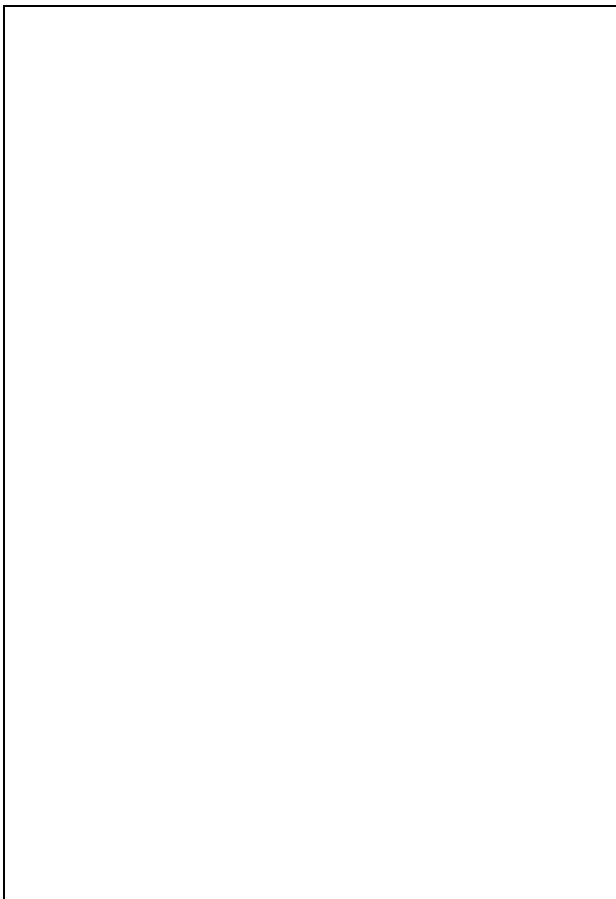
Übernimm das Programm im Bild, führe es aus und zeichne das Ergebnis.

Denk daran, vorher einen neuen Block Dreieck zu programmieren



Aufgabe 3d)

Programmiere das Muster rechts erneut. Nutze dafür jedoch eigene Blöcke, um möglichst viele Programmwiederholungen zu sparen. Schreibe deinen Programmcode in den Kasten links.



Aufgabe 3e)

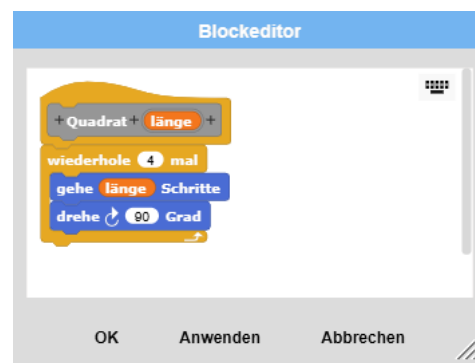
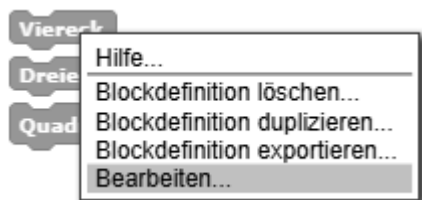
Beschreibe, wie die Blöcke heißen die du erstellt hast und was sie machen.

Aufgabe 3f)

Eigene Blöcke können auch Daten übergeben bekommen.

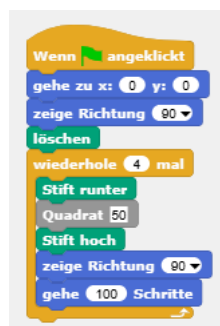
Um den Block zu verändern, öffne durch einen Klick der rechten Maustaste auf ihn das Kontextmenü und klicke dann auf „bearbeiten“ wie links gezeigt.

Klicke danach auf das „+“ rechts neben dem Namen des Blockes und gib einen Namen ein. So heißen die Daten, die in den Block hinein gehen. Du kannst diese Daten im Block verwenden. Man spricht auch von einem „Parameter“.



Aufgabe 3g)

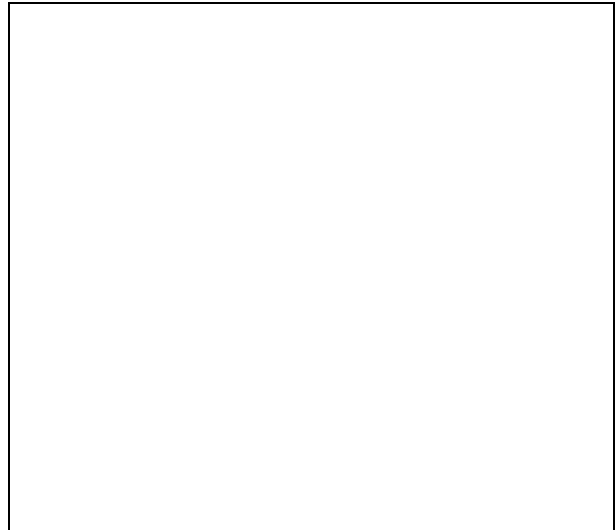
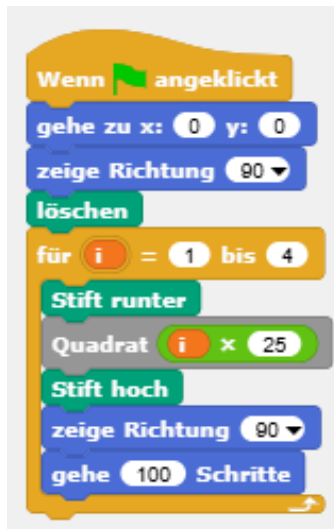
Übernimm das Programm links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 3h)

Bei Wiederholungen kann man auch mitzählen lassen, wie oft die Wiederholung schon ausgeführt wurde. Das kann man benutzen, um jeden Schritt etwas anders zu machen. Übernimm das Programm links, führe es aus und zeichne das Ergebnis rechts ein.

Tipp: Schaue dir den Bereich „Operatoren“ (grün) an.

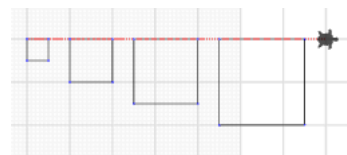
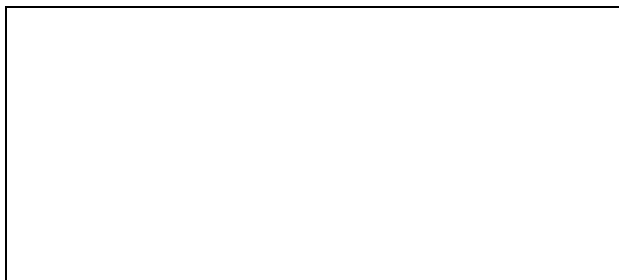


Aufgabe 3i)

Programmiere das Muster rechts, so dass alle Quadrate gleiche Abstände haben.

Schreibe deinen Programmcode in den Kasten links.

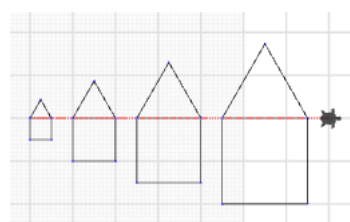
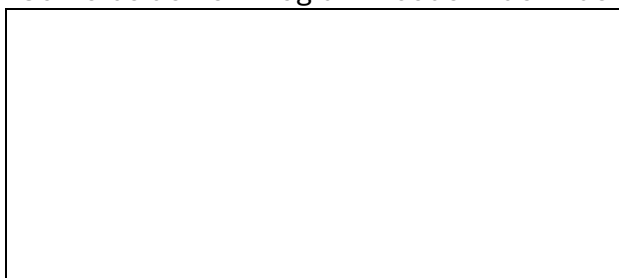
Tipp: Der Abstand muss abhängig von der Größe des Quadrates sein!



Aufgabe 3j)

Programmiere das Muster rechts nach.

Schreibe deinen Programmcode in den Kasten links.

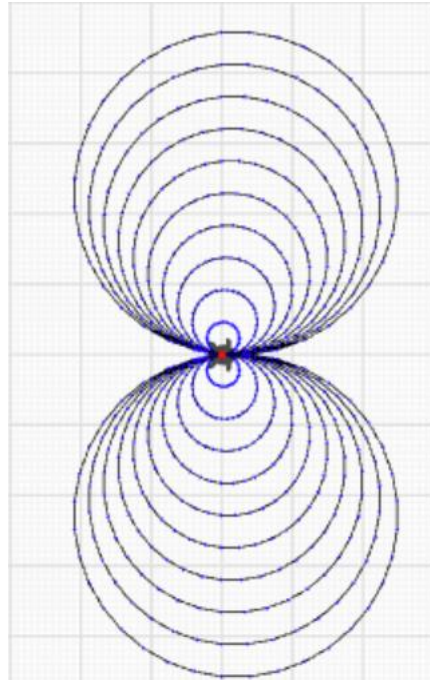
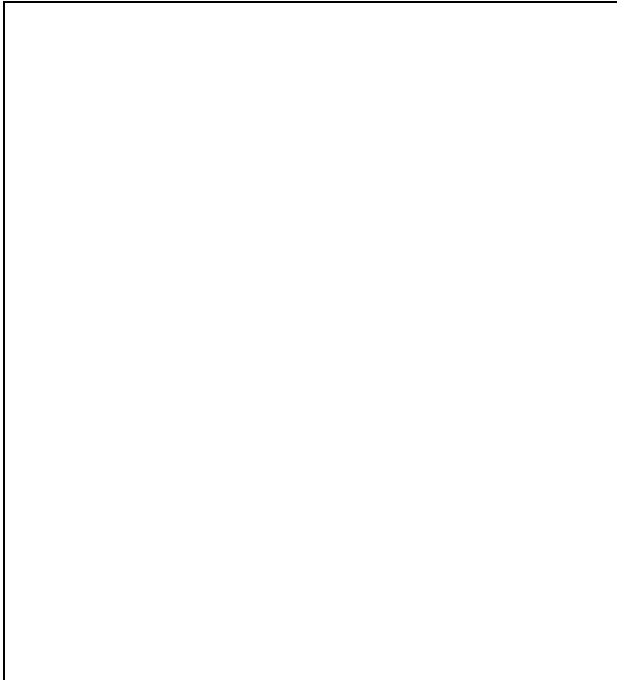


Aufgabe 3k)

Programmiere das Muster rechts nach.

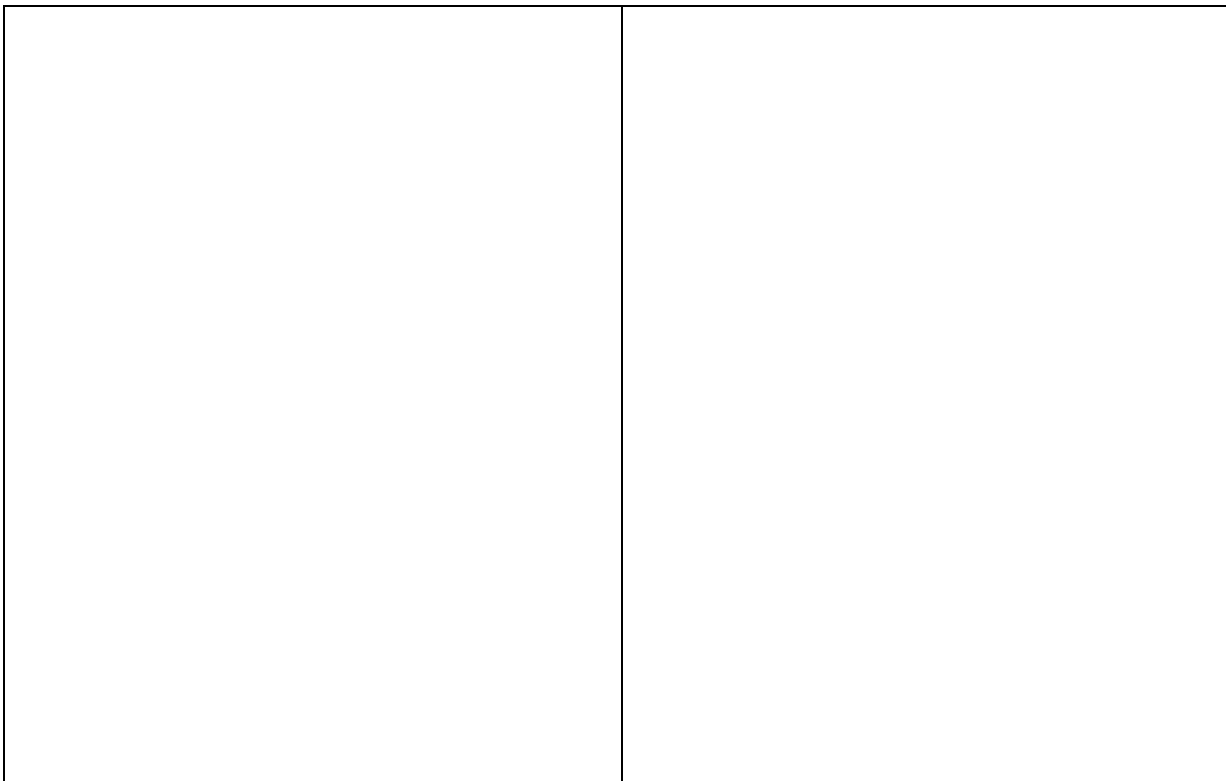
Schreibe deinen Programmcode in den Kasten links.

Hinweis: Es handelt sich hier nicht um Kreise, sondern mehrere 36-Ecken.

**Aufgabe 3l)**

Denke dir ein eigenes Muster aus.

Schreibe deinen Programmcode in den Kasten links und zeichne das Muster in den Kasten rechts.



Aufgabe 4: Fallunterscheidungen

In der Programmierung gibt es auch Fallunterscheidungen. Bevor wir uns mit diesen beschäftigen, müssen wir jedoch zurück in die Grundschule und zum „Teilen mit Rest“. Mit Rest zu teilen ist auch in der Programmierung sehr wichtig. Weil das jedoch nach Grundschule klingt, benutzt man stattdessen einen mathematischen Fachbegriff: modulo. Der Begriff „i modulo 2“ bedeutet: Gebe den Rest aus, den man erhält, wenn man i durch 2 teilt.

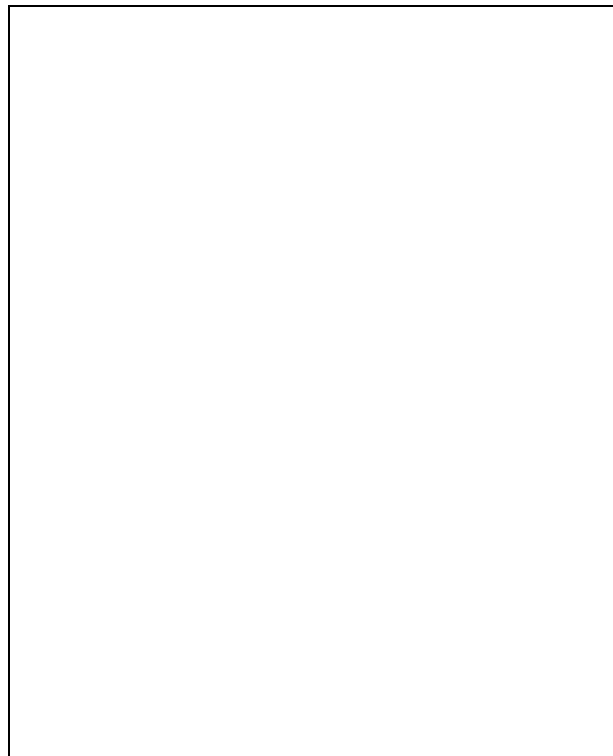
Aufgabe 4a)

Vervollständige die Tabelle.

		
1	0	1
2	1	0
3	1	1
4	2	
5		
6		
7		

Aufgabe 4b)

Übernimm das Programm im Bild links, führe es aus und zeichne das Ergebnis rechts ein. Denk daran, ggf. zuerst die Funktionen Dreieck und Quadrat zu erstellen

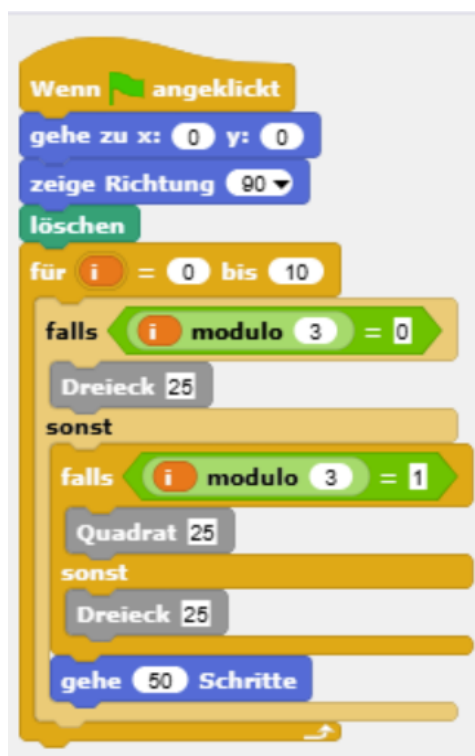


Aufgabe 4c)

Beschreibe in eigenen Worten, was die Zeile „falls (i modulo 2) = 0“ macht.

Aufgabe 4d)

Übernimm das Programm im Bild links, führe es aus und zeichne das Ergebnis rechts ein.



Aufgabe 4e)

Programmiere das Muster rechts nach. Schreibe deinen Programmcode in den Kasten links.

**Aufgabe 4f)**

Programmiere das Muster rechts nach. Schreibe deinen Programmcode in den Kasten links.

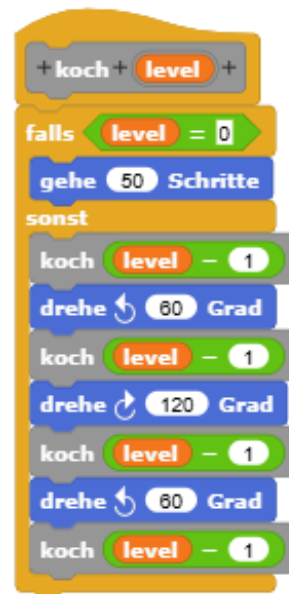


Zusatzaufgabe 5 für Profis: Rekursion

Rekursion bezeichnet eine komplexe Programmiermethode, bei der sich eine Funktion selbst wieder aufruft.

Aufgabe 5a)

Übernimm das folgende Programm und führe es aus.



Aufgabe 5b)

Zeichne das Ergebnis, wenn der Parameter von koch auf 0 (links oben), 1 (rechts oben), 2 (links unten) und 3 (rechts unten) gesetzt wird in die Kästen.

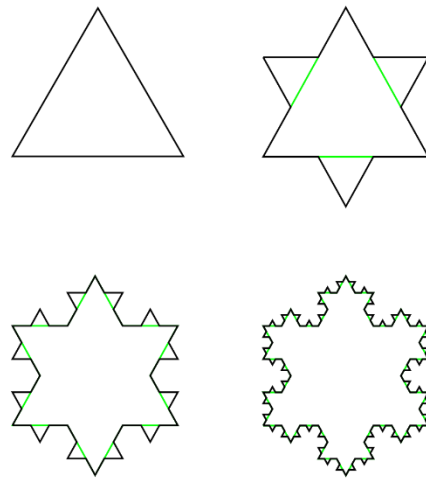
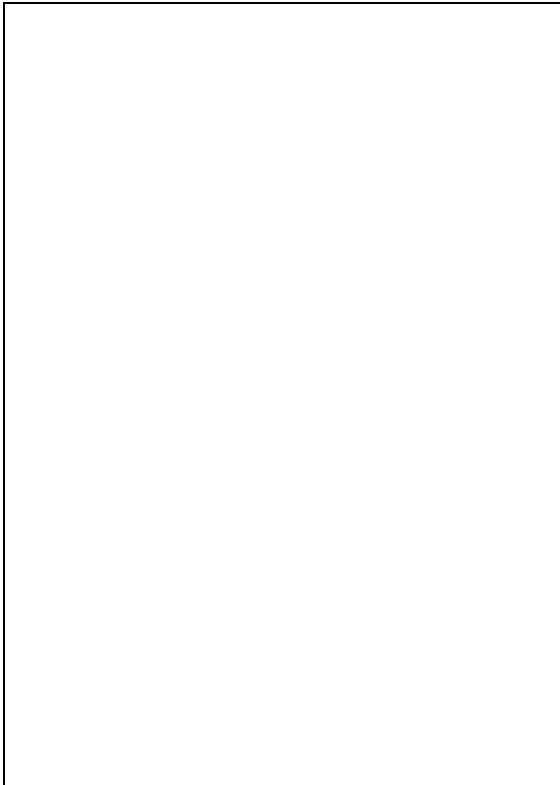
Aufgabe 5c)

Ändere das Programm so ab, dass die Linien nicht größer werden, wenn man den Parameter erhöht.

Tipp: Du benötigst einen zweiten Parameter „Länge“, die bestimmt, wie lang der Schritt ist. Umso höher das Level, umso niedriger muss die Schrittlänge sein

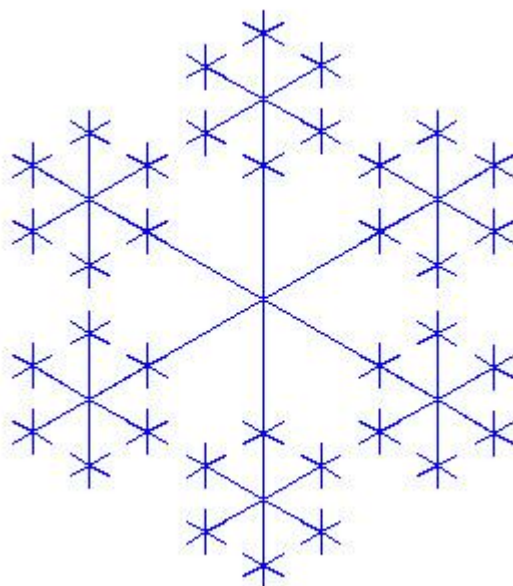
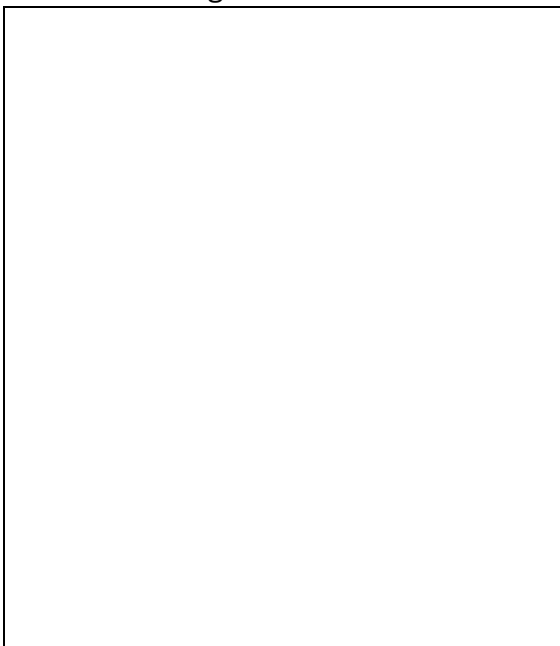
Aufgabe 5d)

Erweitere das Programm, sodass damit die sogenannte „Kochschne Schneeflocke“ gezeichnet wird. Gib den Programmtext der Funktion



Aufgabe 5e)

Zeichne die Figur rechts. Nutze hierfür Rekursion und gib den fertigen Programmtext an.



Aufgabe 6: Putting it all together

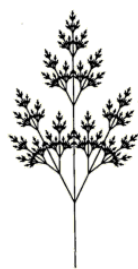
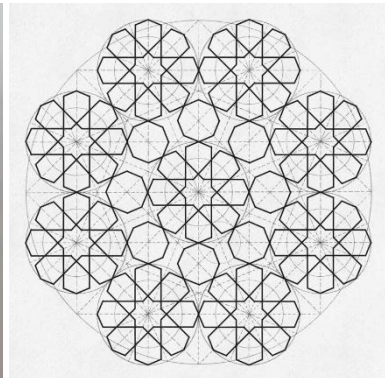
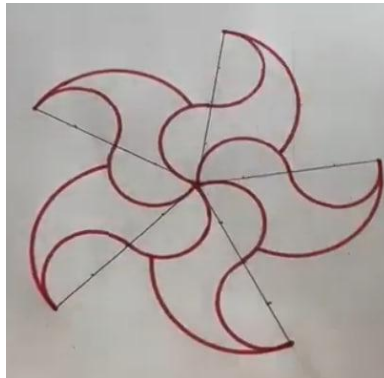
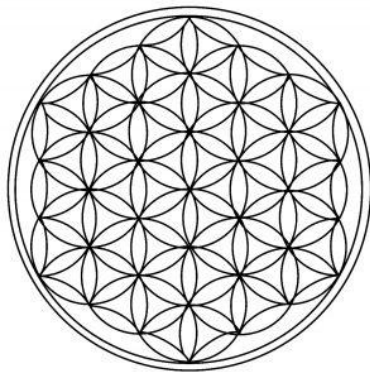
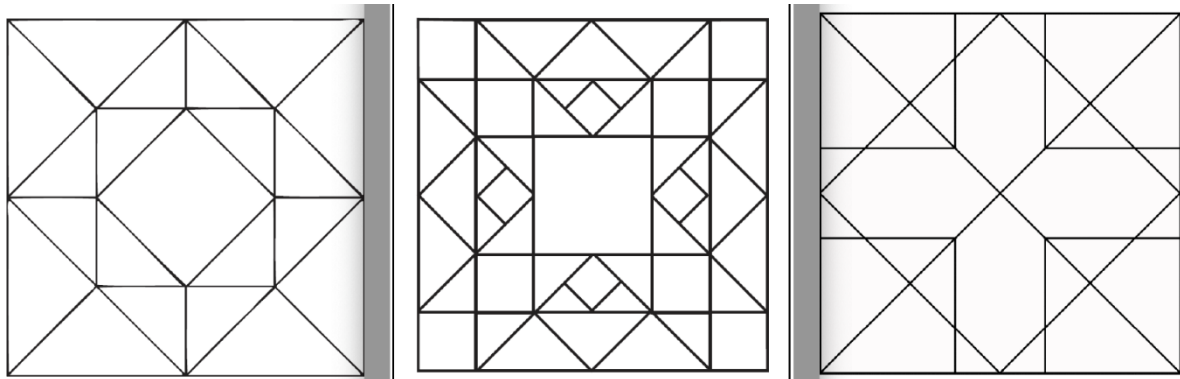
Im den vorherigen Aufgaben hast du verschiedene Techniken kennen gelernt, um Muster zu programmieren. Jetzt ist deine Kreativität gefragt – sowohl beim Design, als auch beim Programmieren!

5a) Erstelle ein schönes Muster.

Damit das Muster gestickt werden kann, muss es folgende Eigenschaften erfüllen:

1. Es darf maximal eine Größe von 20x20cm besitzen
2. Das Muster darf nur eine Farbe verwenden (schwarz, rot, grün, blau)
3. Die Stiche dürfen nicht zu weit und zu nah voneinander entfernt sein. Achte ggf. auf Warnungen im Programm oder lass einen Betreuer drüber schauen.

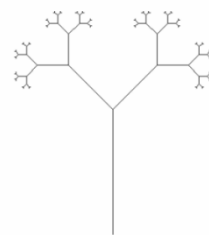
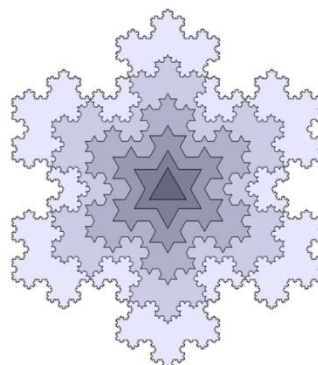
Probiere gerne aus oder lasse dich von Bildern inspirieren:



$n=7, \delta=20^\circ$
 $X \rightarrow F[+X]F[-X]+X$
 $F \rightarrow FF$

$n=7, \delta=25.7^\circ$
 $X \rightarrow F[+X]F[-X]FX$
 $F \rightarrow FF$

$n=5, \delta=22.5^\circ$
 $X \rightarrow F-[X]+X]+F[+FX]-X$
 $F \rightarrow FF$



Speichere das Muster auf deinem Computer und lass es von den Betreuern sticken!