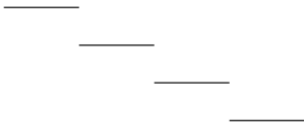


1. (2 Punkte) Skizziere das Turtle-Bild.

```
def zeichne(bremse):
    if bremse == 0: return
    down()
    forward(60)
    up()
    right(90)
    forward(30)
    left(90)
    zeichne(bremse-1)

hideturtle()
zeichne(4)
```

Lösung:



2. (2 Punkte) Skizziere das Turtle-Bild.

```
def zeichne(bremse):
    if bremse == 0: return

    pos = position()
    winkel = heading()

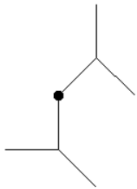
    right(90)
    forward(50)
    zeichne(bremse-1)

    up()
    setpos(pos)
    setheading(winkel)
    down()

    left(45)
    forward(50)
    zeichne(bremse-1)

hideturtle()
dot(10)
zeichne(2)
```

Lösung:



3. (4 Punkte) a. Skizziere das Turtle-Bild.
b. Skizziere das Turtle-Bild, wenn die die Zeile `back(laenge)` auskommentiert ist.

```
def zeichne(laenge, bremse):  
    if bremse == 0: return  
    forward(laenge)  
    back(laenge)  
    right(45)  
    zeichne(laenge*0.5, bremse-1)
```

```
hideturtle()  
zeichne(200,4)
```

Lösung:



a.



b.

4. (3 Punkte) Die Punkte in dem Bild zeigen die Positionen, an denen die rekursive Funktion aufgerufen wird. Schreibe die Reihenfolge der rekursiven Aufrufe an die entsprechenden Punkte.

```
from turtle import *

def zeichne(laenge,bremse):
    if bremse == 0:
        dot(8)
        return
    dot(8)

    pos = position()
    winkel = heading()

    left(40)
    forward(laenge)
    zeichne(laenge*0.7,bremse-1)

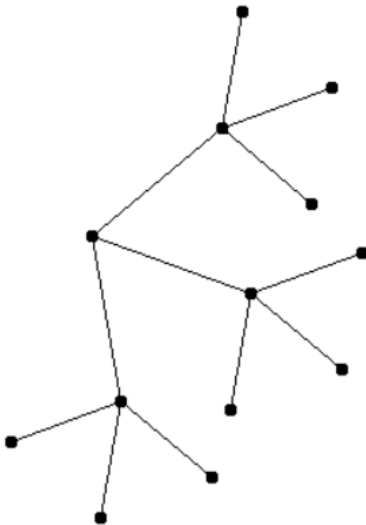
    up()
    setpos(pos)
    setheading(winkel)
    down()

    right(80)
    forward(laenge)
    zeichne(laenge*0.7,bremse-1)

    up()
    setpos(pos)
    setheading(winkel)
    down()

    right(20)
    forward(laenge)
    zeichne(laenge*0.7,bremse-1)

hideturtle()
zeichne(100,2)
```



Lösung:

