Quadratzahlen
Der Sketch rechts berechnet die ersten Zehn Quadratzahlen.

```
for (int i=0; i<=10; i++) {
 int quadrat = i*i;
 println("Das Quadrat von " + i + " lautet " + quadrat);
}
```

Verändere den Sketch so, dass die ersten 100 (*10000, 50000*) Quadratzahlen berechnet werden. Was fällt dir auf?

Finde möglichst genau heraus, ab welcher Zahl das Problem auftritt!

```
int a = 2147483647;

int b = 2147483648;
```

The literal 2147483648 of type int is out of range

1 Wandle um ins Dezimalsystem

a) 1101_2 b) $1\ 0110_2$ c) $1\ 1100\ 0111_2$ d) $1000\ 1000\ 1000_2$

2 Eigenschaften des Binärsystems

Notiere zwei Beispiele zu den Aussagen und vervollständige die Eigenschaft.

a) „Wenn die letzte Ziffer einer Zahl im Binärsystem 1 ist, dann ist die Zahl …"
b) „Ergänzt man eine 0 am Ende einer Binärzahl, dann wird die Zahl …"
„Ergänzt man eine 1 am Ende einer Binärzahl, dann wird die Zahl …"
c) „Ersetze in einer Binärzahl von rechts alle 1en durch 0en bis du eine 0 erreichst. Ersetze diese 0 durch eine 1 und stoppe dann. Die Zahl wird dadurch …"

3 Länge von Binärzahlen

Welches ist die größte und kleinste Zahl, die mit n binären Ziffern dargestellt werden kann? Wie kann die benötigte Anzahl von binären Ziffern für eine Zahl berechnet werden?

4 Umwandlung ins Binärsystem

a) Ergänze die Lücken zur Umwandlung der Zahl 90_{10} in die Binärdarstellung.

1. Schritt: Die Zahl 90 ist gerade. Das letzte Bit in der Binärdarstellung lautet ___.

2. Schritt: Die restlichen Ziffern in der Binärdarstellung bilden die Zahl 90 : 2 = _____.

Diese Zahl ist _____. Das nächste Bit in der Binärdarstellung lautet ___.

3. Schritt: Die restlichen Ziffern in der Binärdarstellung bilden die Zahl (____ - 1) : 2 = _____.

Diese Zahl ist _____. Das nächste Bit in der Binärdarstellung lautet ___.

4. Schritt: Übrige Zahl: _____.
Diese Zahl ist _____. Nächstes Bit: ___.

5. Schritt: Übrige Zahl: _____.
Diese Zahl ist _____. Nächstes Bit: ___.

6. Schritt: Übrige Zahl: _____.
Diese Zahl ist _____. Nächstes Bit: ___.

7. Schritt: Übrige Zahl: _____.
Diese Zahl ist _____. Nächstes Bit: ___.

Lösung: 90_{10} = _____$_2$.

b) Wiederhole das Vorgehen für die Zahl 138_d.

5 Hexadezimalsystem

Die Stellen im Hexadezimalsystem bestehen von rechts nach links aus Vielfachen der Zahlen $16^0 = 1$, $16^1 = 16$, $16^2 = 256$, $16^3 = 4096$, usw. Als jeweilige Vielfache einer Stelle werden zunächst weiterhin die Ziffern 0 bis 9 verwendet. Um 10, 11, 12, 13, 14 oder 15 Einer anzeigen zu können, benötigt man jedoch noch sechs weitere Ziffern. Diese werden in numerisch aufsteigender Reihenfolge mit A, B, C, D, E und F bezeichnet.
Die Zahl $D9_{16}$ bedeutet somit (von rechts nach links) 9 Einer plus 13 Sechzehner, also $9 \cdot 1 + 13 \cdot 16 = 9+208=217$.

Wandle um ins Dezimalsystem:

a) $B5_{16}$ b) $9F_{16}$ c) $AB\ C0_{16}$ d) $AF\ FE_{16}$

Hex	Bin	Dec
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Ein Vorteil am Hexadezimalsystem ist die einfache Umrechnung zwischen dem Binärsystem: Die Binärzahlen können ganz einfach in „Viererpäckchen" unterteilt werden, um die zugehörige Hexadezimalzahl zu erzeugen. Im Beispiel werden aus $1101\ 1001_2$ die beiden Viererpäckchen 1101 und 1001. Jedes für sich als Binärzahl interpretiert ergibt eine Zahl zwischen 0 und 16, hier sind $1101_2 = 13$ und $1001_2 = 9$. Die 13 wird im Hexadezimalsystem aber durch ein D dargestellt und schon hat man die Zahldarstellung $D9_{16}$ erhalten. Das ist deutlich einfacher als die Umrechnung ins Dezimalsystem!

Wandle um ins Hexadezimalsystem:

a) 11101101_2 b) 100010011010_2 c) 1111000_2 d) 11011010111100_2

Wandle um ins Binärsystem:

a) $C3_{16}$ b) 2021_{16} c) $AF\ FE_{16}$

Bei der Umwandlung vom Dezimalsystem ins Hexadezimal kann man wie bei der Umwandlung ins Binärsystem vorgehen. Dabei wird in jedem Schritt eine Division mit Rest durch 16 durchgeführt.

Wandle um ins Hexadezimalsystem:

a) 758_{10} b) 1451_{10} c) 43962_{10}

6 Umwandlungen in Processing

Als Algorithmus zur Umwandlung von Zahlen ins Binärsystem kann am Computer dieselbe Strategie wie von Hand verwendet werden. Dazu benötigen wir eine Division mit Rest.

```
int quotient = zahl / 2;
int rest = zahl % 2;
println(zahl + ":2 = " + quotient + " Rest " + rest);
```

a) Probiere die Division mit Rest an einigen Beispielen mit dem Programmcode links aus.

```
int zahl = 90;

while (_________) {
  int quotient = zahl/2;
  int rest = zahl%2;
  println(zahl + ":2 = " + quotient + " Rest " + rest);
  zahl = _________;
}
```

b) Um die Division mit Rest mehrfach zu wiederholen, wird eine `while`-Schleife eingesetzt. Dabei wird eine Laufbedingung angegeben: Der Schleifenblock wird dann wiederholt, solange die Laufbedingung wahr ist.
Ergänze die Lücken im Programmcode.

c) Wandle mit deinem Programm die Zahl 2147483647 aus dem Einstiegsbeispiel um.

d) Passe das Programm so an, dass eine Umwandlung ins Hexadezimalsystem durchgeführt wird.