

Aufgabe 11

a)

Das grundlegende Verfahren, um drei Integer in einer Speicherzelle zu speichern, sieht so aus, dass man die drei Integer aneinanderreihet.

D.h.: Man konvertiert zunächst das Vorzeichen, eine positive Zahl bekommt eine '0' vorangestellt, eine negative eine '1'. Danach fügt man bei jeweils jeder Zahl zwischen dem Vorzeichen und den Ziffern so viele '0'en ein, dass alle Zahlen die gleiche Ziffernlänge haben. Zum Schluss werden die Zahlen chronologisch aneinandergerreihet.

Als Beispiel: Seien 40, -8 und -712 unsere drei Integer.

Im ersten Schritt wird das Vorzeichen gespeichert: '040', '18', '1712'.

Im zweiten Schritt wird die Ziffernlänge mit '0'en angepasst: '0040', '1008', '1712' ('1712' hat schon die passende Länge).

Wenn man nun alle Ziffern hintereinanderschreibt, hat man den Inhalt der Speicherzelle: '004010081712'.

Die Rekonstruktion der Zahlen geschieht, indem man dieses Verfahren rückwärts abläuft.

b)

Siehe 'Aufgabe11.java' (in Kombination mit RISC_Machine_v11.java)

Anmerkung: Da uns die Problematik mit dem begrenzten Speicher eines Integers bewusst war, haben wir direkt mit 'RISC_Machine_v11.java' ein vollfunktionierendes Programm geschrieben, sodass 'Aufgabe11.java' nur die notwendigen Berechnungen an 'RISC_Machine_v11.java' delegiert.

c)

Was auffällt ist, dass "hinten und vorne nicht dasselbe rauskommt". (siehe main(String[] args) in 'Aufgabe11.java')

Das liegt an dem begrenzten Speicher eines ints mit 32 Bits, sodass nicht drei beliebig große Zahlen in einen Integer geschrieben werden können.

d)

Die Problematik kann man umgehen, indem man variable-precision integers nutzt, also im Fall von Java ist das java.math.BigInteger.

Siehe 'RISC_Machine_v11.java'.