

ADIP Ü2 A1

Christoph Schwerdtfeger

October 27, 2014

1 Aufgabe 1

Output des Programms: ./aufgabe1

Beispiel 1: $35 + 85 = 120$

Beispiel 2: $85 + 85 = -86$

Während das erste Beispiel das korrekte Ergebnis ausgibt, ist das Ergebnis des zweiten Beispiels falsch (im Sinne normaler Arithmetik). Dies liegt daran, dass die Variablen `x1`, `x2`, `result` als `char` initialisiert wurden und deshalb nur eine Wortlänge von einem Byte haben (und deshalb im Zweierkomplement von -128 bis 127 reichen können). 85 entspricht im Zweierkomplement 01010101, deshalb ist $01010101 + 01010101 = 110101010$ - die erste Stelle wird nicht mehr ausgewertet und das Ergebnis ist 10101010, -86 in Dezimalschreibweise.

2 Aufgabe 2

Das erste Verfahren braucht 16 Multiplikationsschritte (abstrahiert von den Fallunterscheidungen zuvor).

Das zweite Verfahren braucht 4 Schritte: Im Exponenten von 17 auf 8, dann 4,2,1.

Beim Exponenten 17 ist der Unterschied nicht so arg groß, weil ja noch die Manipulation der Exponenten und die Modulo-lfs durchgeführt bzw. ausgewertet werden müssen. Bei größeren Exponenten ist der Unterschied erheblich (linear vs log).

3 Aufgabe 3

Die längste Collatzfolge der Zahlen von 1 bis 1000 wird von der 871 erzeugt, sie braucht 171 Schritte. Code beigelegt.