

Blatt 5

Aufgabe 1. 0/6

Aufgabe 2. 0/8

Aufgabe 3.

Geben Sie ein RETI-Programm an, das den logischen Ausdruck $x \leq y$ korrekt zu 0 bzw. 1 auswertet, auch wenn bei der Berechnung von $x - y$ ein Überlauf auftritt.

Lösung.

0	SUBI SP 1;	Stack um eine Zelle erweitern
1	LOAD ACC 10;	Wert von x in ACC laden
2	JUMP \leq 4;	Falls $x \leq 0$, PC um 4 erhöhen (PC = 6)
3	LOAD ACC 11;	Wert von y in ACC laden
4	JUMP \geq 5;	Falls $y \geq 0$ kein Überlauf, PC um 5 erhöhen (PC = 9)
5	JUMP 7;	Ansonsten Überlauf möglich, Ergebnis 0, PC um 7 erhöhen (PC = 12)
6	LOAD ACC 11;	Wert von y in ACC laden
7	JUMP \leq 2;	Falls $y \leq 0$ kein Überlauf, PC um 2 erhöhen (PC = 9)
8	JUMP 6;	Ansonsten Überlauf möglich, Ergebnis 1, PC um 6 erhöhen (PC = 14)
9	LOAD ACC 10;	Wert von x in den ACC laden
10	SUB ACC 11;	Ergebnis von x-y in ACC laden
11	JUMP \leq 3;	Falls $x \leq y$ wahr, PC um 3 erhöhen (PC = 14)
12	LOADI ACC 0;	Ergebnis 0, wenn $x \leq y$ falsch
13	JUMP 2;	PC um 2 erhöhen (PC = 15)
14	LOADI ACC 1;	Ergebnis 1, wenn $x \leq y$ wahr
15	STOREIN SP ACC 1;	Ergebnis auf Stack legen

6/6

Ich hab den RETI-Code aller Studenten die Aufgabe 3 bearbeitet haben (Aufgabe 3 war diesmal die beliebteste Aufgabe) mithilfe des im PicoC-Compilers <https://github.com/matthejue/PicoC-Compiler/releases> eingebauten RETI-Interpreters ausgeführt, genauer mittels des Befehls `picoc_compiler -b -p c.reti -S -P 2 -D 15`. Ich habe versucht den Code von euch Studenten lauffähig zu machen, sodass dieser die Aufgabenstellung erfüllt.

Die Datei <fruit>.in enthält Eingaben für CALL INPUT REG. Die Datei <fruit>.out enthält die Ausgaben der CALL PRINT REG bei der Ausführung. Die Datei <fruit>.out_expected enthält die erwarteten Ausgaben.

Eure Korrektur ist unter https://github.com/matthejue/Abgaben_Blatt_3/tree/main/Blatt5/klementine.reti zu finden.