

Benke Hargitai 5370932  
Lukas Seyfried 5343019

1	2	3	$\Sigma$
0	4	6	

## Aufgabenblatt 05

Abgabe: 25.11.2022

0/6

### Aufgabe 2

#### Teilaufgabe a)

```
st(x) = (var, int, 128) # bds = 128
st(y) = (var, int, 129) # bds+1 = 129
st(z) = (const, int, 5)
```

#### Teilaufgabe b)

...

-4

#### Teilaufgabe c)

Maximale Anzahl an Teilergebnissen bei

$$(x_1 \circ (x_2 \circ (\dots \circ (x_{n-1} \circ x_n))),$$

dabei  $n$  Teilergebnisse vor der ersten schließenden Klammer.

Im Gegensatz dazu sind bei

$$(((x_1 \circ x_2) \circ \dots) \circ x_{n-1}) \circ x_n)$$

die minimale Anzahl von 2 Teilergebnissen ( $x_1$  und  $x_2$ ) notwendig.

### Aufgabe 3

6/6 großartige Arbeit!

Es kann bei der Berechnung  $x - y$  nur zu einem Überlauf kommen, wenn  $x$  und  $y$  unterschiedliche Vorzeichen haben. Diese Fälle fangen wir mit bedingten Sprüngen direkt ab.

```
LOAD ACC 10    ; Lade x in Akkumulator

JUMP> 4         ; Fall 1: x <= 0
LOAD ACC 11    ; Lade y in Akkumulator.
JUMP> 9         ; Falls y > 0: Es gilt x <= 0 < y, also kann x <= y
                  ; ohne weiteren Vergleich zu 1 ausgewertet werden.
JUMP 3          ; Es gilt x <= 0 und y <= 0, weiter mit normaler vglop

LOAD ACC 11    ; Fall 2: Lade y in Akkumulator. Es gilt x > 0
JUMP< 4         ; Falls y < 0: Es gilt x >= 0 > y, also kann x <= y
                  ; ohne weiteren Vergleich zu 0 ausgewertet werden.

LOAD ACC 10    ; ab hier normale vglop aus VL
SUB ACC 11
```

```
JUMP<= 3  
LOADI ACC 0  
  
JUMP 2  
LOADI ACC 1
```

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/kiwi.reti](https://github.com/matthejue/Abgaben_Blatt_3/tree/main/Blatt5/kiwi.reti) zu finden.