

Lösung 3

Aufgabe 16

```
; MUL.LC1

M1:      LDA FAKTOR1 ; A := <FAKTOR1>
        LDB EINS   ; B := 1
        SUB                ; A := A - B
        JPS M99     ; IF A < 0 THEN GOTO M99
        MOV FAKTOR1 ; <FAKTOR1> := A
        LDA PRODUKT ; A := <PRODUKT>
        LDB FAKTOR2 ; B := <FAKTOR2>
        ADD                ; A := A + B
        MOV PRODUKT ; <PRODUKT> := A
        JMP M1       ; GOTO M1
M99:     LDA PRODUKT ; A := <PRODUKT>
        HLT                ; HALT
FAKTOR1: DEF 5        ; FAKTOR1
FAKTOR2: DEF 3        ; FAKTOR2
PRODUKT: DEF 0        ; PRODUKT
EINS:    DEF 1         ; EINS
```

Im Programm MUL.LC1 wird FAKTOR1 als Schleifenzähler "mißbraucht". FAKTOR1 wird solange dekrementiert, bis die Differenz FAKTOR1 - 1 negativ geworden ist. Falls FAKTOR1 negativ ist, bricht das Programm sofort ab.

Eine mögliche Lösung (MUL1.LC1) ist, mit dem Betrag von FAKTOR1 zu rechnen und bei negativem FAKTOR1 das berechnete PRODUKT durch -PRODUKT zu ersetzen. Die **fett** markierten Zeilen sind die dafür notwendigen Ergänzungen und Änderungen.

```
; MUL1.LC1

        LDA FAKTOR1 ; A := <FAKTOR1>
        NOT                ; A := /A, FLAG S SETZEN
        JPS LOOP        ; IF A < 0 GOTO LOOP
        LDB EINS        ; B := 1
        ADD                ; A := A + 1
        MOV FAKTOR1    ; <FAKTOR1> := A
        LDA VORZ        ; A := <VORZ>
        NOT                ; A := /A
        MOV VORZ        ; <VORZ> := A
        ;
M1:     LDA FAKTOR1    ; A := <FAKTOR1>
        LDB EINS        ; B := 1
        SUB                ; A := A - B
        JPS M2        ; IF A < 0 GOTO M2
        MOV FAKTOR1    ; <FAKTOR1> := A
        LDA PRODUKT    ; A := <PRODUKT>
        LDB FAKTOR2    ; B := <FAKTOR2>
        ADD                ; A := A + B
        MOV PRODUKT    ; <PRODUKT> := A
        JMP M1         ; GOTO M1
        ;
M2:   LDA VORZ        ; A := <VORZ>
        NOT                ; A := /A, SF SETZEN
        JPS M99       ; IF SF = 1 GOTO M99
        LDA PRODUKT    ; A := <PRODUKT>
        NOT                ; A := /A
        LDB EINS        ; B := 1
        ADD                ; A := A + B
        MOV PRODUKT    ; <PRODUKT> := A
        ;
M99:    LDA PRODUKT    ; A := <PRODUKT>
        HLT                ; HALT
```

```

FAKTOR1: DEF -5          ; FAKTOR 1
FAKTOR2: DEF -3          ; FAKTOR 2
PRODUKT: DEF 0           ; PRODUKT
EINS:    DEF 1           ; KONSTANTE 1
          ;
VORZ:    DEF 1           ; VORZEICHEN FAKTOR 1

```

Aufgabe 17

Die Aufgabe ist nach Kochrezept "Elementweises Lesen und Verarbeiten eines Arrays" lösbar. MINIMUM ist mit der größten darstellbaren Zahl (511) zu initialisieren. Jede Zahl ist mit MINIMUM zu vergleichen. Falls die Zahl kleiner als das aktuelle MINIMUM ist, ist MINIMUM mit dieser Zahl zu überschreiben. Wenn alle Zahlen verglichen sind, steht in MINIMUM die kleinste der Zahlen. Die fett markierten Zeilen sind aufgabenspezifisch.

```

; MINIMUM.LC1

M1:    LDA ANZ          ; A := <ANZ>
        LDB EINS        ; B := 1
        SUB             ; A := A - B
        MOV ANZ         ; <ANZ> := A
        JPS M99         ; IF A < 0 GOTO M99
M10:   LDA ZAHLEN       ; A := <ZAHLEN>
        ;
        LDB MINIMUM    ; B := <MINIMUM>
        SUB             ; A := A - B
        JPS M11         ; IF SF = 1 GOTO M11
        JPO M12         ; IF OF = 1 GOTO M12
        JMP M20         ; GOTO M20
M11:   JPO M20         ; GOTO M20
M12:   ADD             ; A := A + B
        MOV MINIMUM    ; <MINIMUM> := A
        ;
M20:   LDA M10         ; A := <M10>
        LDB EINS        ; B := 1
        ADD             ; A := A + B
        MOV M10         ; <M10> := A
        JMP M1         ; GOTO M1
M99:   HLT             ; HALT
        ;
EINS:   DEF 1          ; KONSTANTE 1
        ;
MINIMUM: DEF 511      ; ANFANGSWERT FUER MINIMUM
        ;
ANZ:    DEF 10         ; ANZAHL DER ZAHLEN = 10
ZAHLEN: DEF 500
        DEF 400
        DEF -51
        DEF -512
        DEF 200
        DEF 100
        DEF 33
        DEF 44
        DEF 55
        DEF 300

```

Aufgabe 18

Die Aufgabe ist nach Kochrezept "Elementweises Lesen und Verarbeiten eines Arrays" lösbar. Jede Zahl ist mit WERT zu vergleichen. Bei Gleichheit ist GLEICH zu inkrementieren. Wenn alle Zahlen verglichen sind, steht in GLEICH die gesuchte Anzahl. Die **fett** markierten Zeilen sind aufgabenspezifisch.

```

; ENTHALT.LC1

M1:    LDA ANZ      ; A := <ANZ>
      LDB EINS     ; B := 1
      SUB          ; A := A - 1
      MOV ANZ      ; <ANZ> := A
      JPS M99      ; IF A < 0 GOTO M99
M10:   LDA ZAHLEN  ; A := <ZAHLEN>
      ;
      LDB WERT     ; B := <WERT>
      SUB          ; A := A - B
      JPS M20      ; IF SF = 1 GOTO M20
      MAB          ; B := A
      LDA NULL     ; A := 0
      SUB          ; A := A - B
      JPS M20      ; IF SF = 1 GOTO M20
      LDA GLEICH   ; A := <GLEICH>
      LDB EINS     ; B := 1
      ADD          ; A := A + B
      MOV GLEICH   ; <GLEICH> := A
      ;
M20:   LDA M10     ; A := <M10>
      LDB EINS     ; B := 1
      ADD          ; A := A + B
      MOV M10      ; <M10> := A
      JMP M1       ; GOTO M1
M99:   HLT         ; HALT
      ;
NULL:  DEF 0       ; KONSTANTE 0
EINS:  DEF 1       ; KONSTANTE 1
      ;
WERT:  DEF 12      ; X
GLEICH: DEF 0      ; GLEICH
      ;
ANZ:   DEF 10      ; ANZ
ZAHLEN: DEF 200    ; ZAHLEN Z
      DEF 12
      DEF 13
      DEF 12
      DEF 1
      DEF 2
      DEF 3
      DEF -12
      DEF 12
      DEF -2

```

Aufgabe 19

Die Aufgabe ist nach Kochrezept "Elementweises Lesen und Verarbeiten eines Arrays" lösbar. FLAG wird auf 1 (Gleichheit) initialisiert. Die erste Zahl wird in TEMP zwischengespeichert. Danach ist jede Zahl mit TEMP zu vergleichen. Bei der ersten Ungleichheit ist FLAG auf 0 zu setzen und das Programm ist zu beenden. Die **fett** markierten Zeilen sind aufgabenspezifisch.

; GLEICH.LC1

```

      LDA  ZAHLEN ; A := <ZAHLEN>
      MOV  TEMP  ; <TEMP> := A
      ;
M1:   LDA  ANZ    ; A := <ANZ>
      LDB  EINS  ; B := 1
      SUB             ; A := A - B
      MOV  ANZ    ; <ANZ> := A
      JPS  M99    ; IF A < 0 GOTO M99
      ;
M10:  LDA  ZAHLEN ; A := <ZAHLEN>
      ;
      LDB  TEMP  ; B := <TEMP>
      SUB             ; A := A - B
      JPS  M11    ; IF SF = 1 GOTO M11
      LDB  EINS  ; B := 1
      SUB             ; A := A - 1
      JPS  M20    ; IF SF = 1 GOTO M20
M11:  LDA  NULL  ; A := 0
      MOV  FLAG  ; <FLAG> := A
      JMP  M99    ; GOTO M99
      ;
M20:  LDA  M10   ; A := <M10>
      LDB  EINS  ; B := 1
      ADD             ; A := A + B
      MOV  M10   ; <M10> := A
      JMP  M1     ; GOTO M1
M99:  HLT                    ; STOP
      ;
NULL:  DEF  0      ; KONSTANTE 0
EINS:  DEF  1      ; KONSTANTE 1
      ;
TEMP: DEF  0      ; TEMP
FLAG: DEF  1      ; FLAG
      ;
ANZ:   DEF  5      ; ANZAHL
ZAHLEN: DEF  5      ; ZAHLENFELD
      DEF  5      ;
      DEF  5      ;
      DEF  5      ;
      DEF  6      ;
```