

# Vorlesungsaufgaben

*(Vorlesungsaufgaben)***Stichwörter:** Ein-Adress-Befehl-Assembler

## Vorlesungsaufgaben

Geben Sie die Lösungen zu den Aufgaben aus der Assembler-Vorlesung ab. Bearbeiten Sie erst danach die folgenden Aufgaben auf diesem Übungsblatt.

(a) Folie 28/2: Berechnung der Potenz  $a^n$ .

### Assembler

```
# WHILE n <> 0 DO
solange:      LOAD n
              JMPP rechnung
              JMP ende

# ergebnis := ergebnis * a;
rechnung:     LOAD a
              MUL ergebnis
              STORE ergebnis

# n := n - 1;
              LOAD n
              SUBI 1
              STORE n
              JMP solange

ende:         HOLD

# a := 2;
a:           WORD 2
# n := 8;
n:           WORD 8
# ergebnis := 1;
ergebnis:     WORD 1
```

### Minisprache

```
PROGRAM potenz;
VAR a, n, ergebnis;
BEGIN
  a := 2;
  n := 8;
  ergebnis := 1;
  WHILE n <> 0 DO
    ergebnis := ergebnis * a;
    n := n - 1;
  END
END potenz.
```

## (b) Folie 28/3: Größten gemeinsamen Teiler zweier Zahlen

**Assembler**

```
# a := 10;
                                LOADI 3
                                STORE a

# b := 5;
                                LOADI 30
                                STORE b

# IF a = 0 THEN
                                LOAD a
                                CMPI 0
                                JMPNZ solange

# ggt := b;
                                LOAD b
                                STORE ggt
                                JMP ende

# WHILE b <> 0 DO
solange:                       LOAD b
                                CMPI 0
                                JMPZ setze_ergebnis

# IF a > b THEN
                                LOAD a
                                CMP b
                                JMPNP setze_b

# a := a - b;
                                LOAD a
                                SUB b
                                STORE a
                                JMP solange

# b := b - a;
setze_b:                       LOAD b
                                SUB a
                                STORE b
                                JMP solange

# ggt := a;
setze_ergebnis:               LOAD a
                                STORE ggt

ende:                           HOLD

a:                               WORD 0
b:                               WORD 0
ggt:                            WORD 0
```

**Minisprache**

```
PROGRAM ggt;
VAR a, b, ggt;

BEGIN
  a := 10;
  b := 5;
  IF a = 0 THEN
    ggt := b;
  ELSE
    WHILE b <> 0 DO
      IF a > b THEN
        a := a - b;
      ELSE
        b := b - a;
      END;
    END;
    ggt := a;
  END
END ggt.
```



## Die Bschlangaul-Sammlung Hermine Bschlangauland Friends

Eine freie Aufgabensammlung mit Lösungen von Studierenden für Studierende zur Vorbereitung auf die 1. Staatsexamensprüfungen des Lehramts Informatik in Bayern.



Diese Materialsammlung unterliegt den Bestimmungen der Creative Commons Namensnennung-Nicht kommerziell-Share Alike 4.0 International-Lizenz.

Hilf mit! Die Hermine schafft das nicht allein! Das ist ein Community-Projekt! Verbesserungsvorschläge, Fehlerkorrekturen, weitere Lösungen sind herzlich willkommen - egal wie - per Pull-Request oder per E-Mail an [hermine.bschlangaul@gmx.net](mailto:hermine.bschlangaul@gmx.net). Der TeX-Quelltext dieses Dokuments kann unter folgender URL aufgerufen werden: [https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50\\_TECH/10\\_Ein-Adress/Aufgabe\\_01-Vorlesungsaufgaben.tex](https://github.com/bschlangaul-sammlung/examens-aufgaben/blob/main/Module/50_TECH/10_Ein-Adress/Aufgabe_01-Vorlesungsaufgaben.tex)