



Arbeitsblatt 5.3

Thema: Programmieren mit MOPS — Anspruchsvoll (Algorithmenideen)

Bearbeitungshinweise

- **Arbeitsform:** Gruppenarbeit (2–3 Personen) für die Aufgaben 1–4; Einzelarbeit/Hausaufgabe für die Aufgaben 5–9.
- **Abgabe:** Gruppen: kurzer Code-Screenshot oder Datei des MOPS-Programms mit 1–2 Stichpunkten zur Idee. Hausaufgaben: bis zur nächsten Stunde.
- **Testen:** Nutzt die angegebenen Testfälle und ergänzt 1–2 eigene Randfälle.
- **MOPS-Kurzreferenz:** `in`, `out`, `ld`, `st`, `add`, `sub`, `mul`, `div`, `mod`, `cmp`, `jmp`, `jlt`, `jeq`, `jgt`, `end`. Eine Anweisung je Zeile; Sprungmarken nach dem Befehl definieren.

Ziel

Ihr setzt **anspruchsvollere Algorithmen** im **MOPS-Befehlssatz** um (Schleifen, Verzweigungen, Invarianten) und achtet auf korrekte Abbruchbedingungen sowie Sonderfälle.

Gruppenauftrag

Aufgabe 1: Euklidischer ggT.

[10 BE]

I/O: Lies a , b und gib den $\text{ggT}(a, b)$ aus.

Idee: Solange $b \neq 0$: $t = a \bmod b$; $a = b$; $b = t$. Am Ende ist a der ggT.

Erweiterung (*): Zusätzlich kgV via $\text{kgV}(a_0, b_0) = \frac{a_0 \cdot b_0}{\text{ggT}(a_0, b_0)}$ (mit ursprünglichen Werten).

Tests: $(48, 18) \rightarrow 6$ · $(21, 14) \rightarrow 7$ · $(10, 0) \rightarrow 10$.

Aufgabe 2: Primtest (Trial Division).

[10 BE]

I/O: Lies n und gib 1, falls n prim ist, sonst 0.

Idee: Prüfer i von 2 aufwärts; solange $i \cdot i \leq n$: wenn $n \bmod i = 0$, dann nicht prim.

Spezialfälle: $n < 2 \rightarrow 0$, $n = 2 \rightarrow 1$.

Tests: $1 \rightarrow 0$ · $2 \rightarrow 1$ · $17 \rightarrow 1$ · $21 \rightarrow 0$.

Aufgabe 3: Binärdarstellung.

[8 BE]

I/O: Lies n und gib die Bits von *LSB nach MSB* aus.

Idee: Wiederholt $n \bmod 2$ ausgeben und $n \div 2$ ausführen, bis $n = 0$; optional Puffer für MSB→LSB.

Tests: $6 \rightarrow 0, 1, 1$ (LSB→MSB) · $13 \rightarrow 1, 0, 1, 1$.

Aufgabe 4: Linearer Suchlauf in kleinem Feld.

[10 BE]

I/O: Lies 5 Werte sowie einen Suchschlüssel **key**; gib den **Index** (0..4) des ersten Treffers aus, sonst -1.

Idee: Werte in feste Zellen laden (z. B. `v0`..`v4`); Zählschleife über Indizes, Vergleich mit **key**.

Tests: $[4, 8, 5, 8, 2], \text{key} = 8 \rightarrow 1$ · $[3, 3, 3, 3, 3], \text{key} = 7 \rightarrow -1$.

Hausaufgaben / Vertiefung

Aufgabe 1: Kleiner Taschenrechner (+, −, *, /).

[8 BE]

I/O: Lies `op` (1..4), `x`, `y` und gib das Ergebnis aus (1:+, 2:−, 3:*, 4:/).

Idee: `cmp op` und passend verzweigen; Division ganzzahlig, bei $y = 0$ z. B. 0 ausgeben.

Tests: $(1, 7, 5) \rightarrow 12$ · $(2, 7, 5) \rightarrow 2$ · $(3, 7, 5) \rightarrow 35$ · $(4, 7, 5) \rightarrow 1$.

Aufgabe 2: Fibonacci mit Limit.

[8 BE]

I/O: Lies `start1`, `start2`, `limit`; gib die Folge bis $\leq \text{limit}$.

Idee: Startwerte ausgeben; dann immer $\text{next} = a + b$ bilden und ausgeben, solange $\text{next} \leq \text{limit}$.

Tests: $(1, 1, 20) \rightarrow 1, 1, 2, 3, 5, 8, 13$ · $(2, 3, 25) \rightarrow 2, 3, 5, 8, 13, 21$.

Aufgabe 3: Zinseszins (ganzzahlig).

[10 BE]

I/O: Lies Kapital K , Zinssatz p (in Promille, also pro 1000), Jahre n ; gib den Endwert aus.

Idee: n -mal: $K \leftarrow K + \lfloor K \cdot p/1000 \rfloor$ (ganzzahlig).

Tests: $(K, p, n) = (1000, 50, 2) \rightarrow 1102$ · $(200, 25, 3) \rightarrow 215$.

Aufgabe 4: Collatz-Folge.

[8 BE]

I/O: Lies n und gib die Folge bis 1 aus.

Idee: Wenn n gerade, dann $n \leftarrow n/2$, sonst $n \leftarrow 3n + 1$; jede Zwischenzahl ausgeben.

Tests: $n = 6 \rightarrow 6, 3, 10, 5, 16, 8, 4, 2, 1$.

Aufgabe 5: Dreieckszahlen / Summenformel prüfen.

[8 BE]

I/O: Lies n ; berechne $S = 1 + \dots + n$ per Schleife und vergleiche mit $n(n+1)/2$.

Idee: Beide Werte ausgeben (z. B. `S` und `Formel`); optional nur 1/0 für Gleichheit.

Tests: $n = 1 \rightarrow S = 1, F = 1$ · $n = 5 \rightarrow S = 15, F = 15$ · $n = 10 \rightarrow S = 55, F = 55$.