

---

## Übungsblatt 3

**Abgabe der MIPS-Assembler-Programme:** Die Lösungen zu den Übungsaufgaben 3.2 – 3.6 sind vor Ende der Kreuzelfrist via Moodle abzugeben (nur für angekreuzte Aufgaben). Packen Sie dafür den MIPS-Assembler-Code als Textdateien, die mit MARS ausführbar sind, in eine einzige ZIP-Datei.

Achten Sie bei allen Assembler-Programmen darauf, dass die MIPS-Prozeduraufrufskonventionen eingehalten werden!

### Ü 3.1 MIPS-Prozeduraufrufe – Grundlagen

---

Machen Sie sich mit den MIPS-Prozeduraufrufskonventionen vertraut, und beantworten Sie folgende Fragen:

- (a) Wie erfolgt die Parameterübergabe an Prozeduren? Welche Rolle spielen dabei Register und Stack?
- (b) Was versteht man unter einem *procedure call frame*? Unter welchen Umständen kann darauf verzichtet werden?
- (c) Worin besteht der Unterschied zwischen *caller-saved* bzw. *callee-saved* Registern? Erklären Sie anhand der Konventionen den Gebrauch der Register \$v0, \$a0, \$t2, \$sp, \$fp, \$ra und \$s0.

### Ü 3.2 MIPS – Funktion mit einem Parameter

---

Implementieren Sie die Funktion  $f(n) = n!$  (Fakultät<sup>1</sup> einer ganzen Zahl  $n \geq 0$ ) als (nicht rekursive) MIPS-Assembler-Funktion, und schreiben Sie ein MIPS-Programm, das die Funktionswerte für  $n = 0, 1, \dots, 10$  am Bildschirm ausgibt! (Jeder Wert soll in einer neuen Zeile erscheinen.)

### Ü 3.3 MIPS – Funktion mit zwei Parametern

---

Schreiben Sie eine MIPS-Assembler-Funktion, die den Binomialkoeffizienten<sup>2</sup>  $b(n, k)$  nach einer der unten angeführten Formeln berechnet. Falls nötig, können Sie die Funktion aus Übungsaufgabe 3.2 übernehmen und aufrufen. Erstellen Sie ein MIPS-Programm, das die ganzzahligen Parameter  $n$  und  $k$  von der Tastatur einliest (Sie dürfen annehmen, dass  $0 \leq k \leq n$  gilt),  $b(n, k)$  durch Funktionsaufruf berechnet und das Ergebnis am Bildschirm ausgibt!

$$b(n, k) = \binom{n}{k} = \frac{n!}{k!(n-k)!} = \prod_{i=1}^k \frac{n-k+i}{i}$$

---

<sup>1</sup> [https://de.wikipedia.org/wiki/Fakult%C3%A4t\\_\(Mathematik\)](https://de.wikipedia.org/wiki/Fakult%C3%A4t_(Mathematik))

<sup>2</sup> <https://de.wikipedia.org/wiki/Binomialkoeffizient>

**Hinweis:** Falls Sie die letzte Formel verwenden, achten Sie bei der iterativen Produktbildung darauf, die Multiplikation jeweils vor der Division auszuführen, um stets ganzzahlige Quotienten zu gewährleisten.

### Ü 3.4 MIPS – Einfach rekursive Funktion

---

Schreiben Sie eine rekursive MIPS-Assembler-Funktion, die die Fakultätsfunktion  $f(n)$  aus Übungsaufgabe 3.2 implementiert. Übernehmen Sie das Hauptprogramm von derselben Aufgabe, um Ihre Implementierung zu testen.

**Hinweis:** es gilt  $f(n) = n * f(n-1)$  für  $n > 0$  und  $f(0) = 1$ .

### Ü 3.5 MIPS – Doppelt rekursive Funktion mit einem Parameter

---

Schreiben Sie eine MIPS-Assembler-Funktion, die die  $n$ -te Fibonacci-Zahl  $F(n)$  durch Rekursion berechnet ( $n \geq 1$  und ganzzahlig). Die Fibonacci-Zahlen sind wie folgt definiert<sup>3</sup>:

$$\begin{aligned} F(1) &= F(2) = 1 \\ F(n) &= F(n-1) + F(n-2) \quad \text{für } n > 2 \end{aligned}$$

Übernehmen Sie das Hauptprogramm aus Übungsaufgabe 3.2, um die ersten zehn Fibonacci-Zahlen ( $n = 1, 2, \dots, 10$ ) am Bildschirm auszugeben.

### Ü 3.6 MIPS – Doppelt rekursive Funktion mit zwei Parametern

---

Schreiben Sie eine rekursive MIPS-Assembler-Funktion zur Berechnung des Binomialkoeffizienten  $b(n, k)$  (vgl. Übungsaufgabe 3.3). Übernehmen Sie das Hauptprogramm aus Übungsaufgabe 3.3, um die Funktion zu testen.

**Hinweis:**

$$\begin{aligned} b(n, k) &= b(n-1, k-1) + b(n-1, k) \quad \text{für } 0 < k \leq n, \\ b(n, 0) &= b(n, n) = 1 \quad \text{für } n \geq 0. \end{aligned}$$

---

<sup>3</sup> <http://de.wikipedia.org/wiki/Fibonacci-Folge>