

---

**Compilerbau**

---

Blatt 11

Abgabe: 30.1.2002

1. [15 Punkte] Schreiben Sie eine endrekursive Fassung von `fib` in POWER-Assembler. Platzieren Sie dabei Ihren Code in folgendes Skelett:

```
.csect .text[PR]
        .globl .fib
.fib:
        # your code goes here ...
        br
```

Das Skelett findet sich auch in der Datei `power/fib.s` und kann mit

```
gcc -o fib fibmain.c fib.s
```

zu einem ausführbaren Programm `fib` gemacht werden. (`fibmain` ruft `fib` mit dem Argument 7 auf.)

Dabei ist das Argument zu `fib` in GPR 3, und `fib` sollte auch den Rückgabewert in GPR 3 hinterlassen. Zur freien Verwendung stehen die GPR's 3–12 zur Verfügung. Der Rücksprung erfolgt durch einen Branch zum Inhalt von LR beim Funktionseintritt.

Im Falle eines arithmetischen Überlaufs sollte `fib` den Rückgabewert -1 liefern.

2. [10 Punkte] Schreiben Sie ein Stück Assemblercode, das in einer Zeichenkette eine Teilkette findet. Die Teilkette soll dabei in GPR 4 als Deskriptor übergeben werden, die Zeichenkette selbst in GPR 3. Der Code soll das Ergebnis in GPR 3 als Deskriptor für einen booleschen Wert zurückgeben. Schreiben Sie ein kleines Rahmenprogramm und überprüfen Sie die Korrektheit des Codes anhand von repräsentativen Beispielen.
3. [5 Punkte] Schreiben Sie eine Funktion, welche die `CChar`- und `CInt`-Literele des Typs `Lambda.const` in die entsprechende Maschinenrepräsentation mit Tags umrechnet. Testen Sie die Ausgabe, insbesondere für Grenzfälle. Lösen Sie eventuell auftretende Probleme, ggf. nach einem Blick in das Objective-Caml-Handbuch!