
ÜBUNG 2

Ziel dieser Übung ist es, die Ein- und Ausgabe mit `scanf` und `printf` zu üben, die Steuerung des Programmflusses mit Hilfe von Schleifen und Verzweigungen zu erlernen und das zentrale Konzept von Funktionen und Modularisierung zu erfassen. Die Übung bezieht sich dabei vorwiegend auf Inhalte der Vorlesung vom 9. Dezember 2013. Die Übung soll im Selbststudium in den Pools durchgeführt werden. Tutoren stehen zu bestimmten Übungsterminen begleitend zur Verfügung.

1. Kopieren Sie die Datei „`converter.c`“ aus dem Verzeichnis `/home/rlehre/EIP1314` und Implementieren Sie die fehlenden Funktionsdefinitionen entsprechend den Kommentaren in der Datei. Kompilieren und Testen Sie das Programm.
10. Die Fibonacci-Folge ist eine unendliche Folge von Zahlen (den Fibonacci-Zahlen), bei der sich die jeweils folgende Zahl durch Addition ihrer beiden vorherigen Zahlen ergibt: 0, 1, 1, 2, 3, 5, 8, 13, ... Benannt ist sie nach Leonardo Fibonacci, der damit 1202 das Wachstum einer Kaninchenpopulation beschrieb. Das rekursive Bildungsgesetz ist wie folgt definiert:

$$f_n = f_{n-1} + f_{n-2} \quad \text{für } n \geq 2$$
$$f_0 = 0 \text{ und } f_1 = 1$$

11. Erstellen Sie ein neues C-Programm `fibonacci.c`.
12. Implementieren Sie eine rekursive Funktion „`long fib_rec(long n)`“, welche f_n berechnet.
13. Implementieren Sie eine nicht-rekursive Funktion „`long fib_iter(long n)`“, welche f_n berechnet. Nutzen Sie dazu eine Schleife Ihrer Wahl.
14. Erstellen Sie eine `main()`-Funktion und geben Sie dort mit Hilfe beider Funktionen die Fibonacci-Zahlen von 1 bis 100 aus. Verifizieren Sie Ihr Ergebnis mit $f_{42} = 267914296$.
15. Erstellen Sie eine Datei „`fibonacci.h`“ im Verzeichnis von „`fibonacci.c`“ und *deklariieren* Sie die beiden Funktionen aus Aufgabe 12. Und 13. Kommentieren Sie die gesamte `main()`-Funktion in „`fibonacci.c`“ aus.
16. Übersetzen Sie die Datei `fibonacci.c` in Maschinsprache in eine Datei „`fibonacci.o`“. Benutzen Sie dafür die Option `-c` des `gcc`-Compilers.
17. Erstellen Sie eine neue Datei „`main.c`“ im selben Verzeichnis, binden Sie Ihre Interfacedeklaration aus „`fibonacci.h`“ ein. Kopieren Sie die auskommentierte `main()`-Funktion aus „`fibonacci.c`“ nach `main.c`. Entfernen Sie die Kommentare.
18. Erstellen Sie ein ausführbares Binärprogramm „`main`“ mit Hilfe des `gcc` aus der Quellcodedatei `main.c` und der Maschinencode-Datei „`fibonacci.o`“.
19. Führen Sie das Programm aus und überprüfen Sie die Ausgaben.
20. Sie haben soeben Ihre erste C-Programmbibliothek geschrieben und eingesetzt!