



Aufgabenblatt 4

Ausgabe: 21.10.2022 14:00

Abgabe: 25.10.2022 22:00

Thema: Rekursion

Abgabemodalitäten

1. Die Aufgaben des C-Kurses bauen aufeinander auf. Versuche daher die Aufgaben zeitnah zu bearbeiten.
2. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf Deinem Rechner mit dem Befehl `clang -std=c11 -Wall -g` kompilieren.
3. Die Abgabe für den Quellcode erfolgt ausschließlich über unser Git im entsprechenden Branch. Nur wenn ein Ergebnis im [ISIS-Kurs](#) angezeigt wird, ist sichergestellt, dass die Abgabe erfolgt ist. Die Abgabe ist bestanden, wenn Du an Deinem Test einen grünen Haken siehst.
4. Du kannst bis zur Abgabefrist beliebig oft neue Versionen abgeben. Lies Dir die Hinweise der Tests genau durch, denn diese helfen Dir, Deine Abgabe zu korrigieren.
Bitte beachte, dass ausschließlich die letzte Abgabe gewertet wird.
5. Die Abgabe erfolgt, sofern nicht anders angegeben, in folgendem Branch: `ckurs-b<xx>-a<yy>`, wobei `<xx>` durch die zweistellige Nummer des Aufgabenblattes und `<yy>` durch die entsprechende Nummer der Aufgabe zu ersetzen sind.
6. Gib für jede Aufgabe die Quellcodedatei(en) gemäß der Vorgabe ab. Im [ISIS-Kurs](#) werden zum Teil Vorgabedateien bereitgestellt. Nutze diese zur Lösung der Aufgaben.
7. Die Abgabefristen werden vom Server überwacht. Versuche, Deine Abgabe so früh wie möglich zu bearbeiten. Du minimierst so auch das Risiko, die Abgabefrist auf Grund von „technischen Schwierigkeiten“ zu versäumen. Eine Programmieraufgabe gilt als bestanden, wenn alle bewerteten Teilaufgaben bestanden sind.

Die Fibonacci-Folge

Vielleicht kennst Du die Fibonacci-Folge schon aus der Schule. Ansonsten empfehlen wir Dir, den Eintrag auf Wikipedia^a anzuschauen. Jede Zahl dieser Folge ist als Summe der beiden vorhergehenden Zahlen definiert. Nur die ersten beiden Fibonacci-Zahlen sind fix auf 1 definiert. Die ersten fünf Fibonacci-Zahlen sind dementsprechend: 1, 1, 2, 3, 5. Die darauf folgende Zahl wäre 3 + 5, also 8. Diese Folge hat kein Ende, sondern läuft unendlich weiter. Mathematisch kann diese Reihe wie folgt ausgedrückt werden:

$$\text{fib}(x) = \begin{cases} x = 1 : 1 \\ x = 2 : 1 \\ x > 2 : \text{fib}(x-1) + \text{fib}(x-2) \end{cases}$$

Die Fibonacci-Folge kann mithilfe von Schleifen (also iterativ) gelöst werden. Ein Code-Beispiel folgt:

```
int n = 10;
int zahla = 1;
int zahlb = 1;
int temp;

printf("Fib(1) = 1\n");
printf("Fib(2) = 1\n");
for (int i=3; i <=n; i++) {
    temp = zahla + zahlb;
    printf("Fib(%d) = %d + %d = %d\n", i, zahla, zahlb, temp);
    zahlb = zahla;
    zahla = temp;
}
```

^asiehe <https://de.wikipedia.org/wiki/Fibonacci-Folge>

Aufgabe 1 Fibonacci

Schreibe ein Programm, das rekursiv die n 'te Fibonacci-Zahl der Fibonacci-Folge ausgibt. Der Nutzer gibt über das Terminal die Zahl n ein, und das Programm gibt dann die n 'te Fibonacci-Zahl zurück. Ein beispielhafter Aufruf inklusive der Ausgabe wird in Listing 1 gezeigt.

Listing 1: Programmbeispiel

```
1 > clang -std=c11 -Wall -g ckurs_blat04_aufgabe01.c input.c
2 -o ckurs_blat04_aufgabe01
3 > ./ckurs_blat04_aufgabe01
4 Bitte gib eine Nummer ein: 4
5 Fib(4) = 3
6
7 > ./ckurs_blat04_aufgabe01
8 Bitte gib eine Nummer ein: 6
```

9 `Fib(6) = 8`

Wie Du sehen kannst, kompilieren wir in Listing 1 die Aufgabe mit der Datei `input.c`. Wir benutzen `input.c` als Bibliothek, um die Funktion `lese_int()` in unser Programm einzubinden. Diese praktische Funktion ermöglicht es, den Benutzer während der Ausführung des Programms um die Eingabe einer Zahl zu bitten. Zu diesem Zweck müssen alle drei Dateien im selben Ordner liegen.

Woher weiß C, dass es diese Funktion gibt? Das kommt daher, dass in beiden Dateien `ckurs_blatt04_aufgabe01.c` und `input.c` eine dritte Datei `input.h` eingebunden ist. Diese "Header"-Datei fungiert als Schnittstelle und definiert die Funktion `lese_int`.

Um die Hausaufgabe zu vereinfachen, bitten wir Dich, die vorgegebene Programmstruktur zu verwenden (siehe Listing 2). Du findest die Vorgabe auch in unserem [ISIS-Kurs](#). Die Abgabe muss den folgenden Kriterien entsprechen:

- Die Zahl wird mithilfe von `lese_int()` eingelesen.
- Die zusätzliche Ausgabe beschränkt sich auf eine Zeile.
- Es werden keine Schleifen verwendet.
- Die gesuchte Zahl wird wie in der Vorlage ausgegeben.

Hinweis: Da wir den Datentyp `int` verwenden, können nur die Fibonacci-Zahlen bis einschließlich `Fib(23)` (also 28657) berechnet werden.

Listing 2: Vorgabe Programmstruktur

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "input.h" // Hier binden wir die Bibliothek ein
4
5 // Schreibe hier die Funktion "int fibonacci"
6
7 int main(){
8     int n = lese_int(); // Hier rufen wir die Funktion lese_int auf.
9     int f = fibonacci(n);
10    printf("Fib(%d) = %d\n" , n, f);
11
12    return 0; //Beende das Programm ohne Fehlermeldung
13 }
```

Nutze zur Lösung der Aufgabe die Vorgaben aus unserem [ISIS-Kurs](#). Füge Deine Lösung als Datei `ckurs_blatt04_aufgabe01.c` im entsprechenden Abgabebereich in Dein persönliches Repository ein und übertrage die Lösung an die Abgabepattform.