



Institut für Angewandte Informationsverarbeitung

Dr. Andreas F. Borchert
Fabian Berstecher

27. Oktober 2017
Blatt 2

Systemnahe Software I (WS 2017/2018)

Abgabe bis zum 3. November 2017, 14:00 Uhr

Lernziele:

- Grundlegende Kontrollstrukturen in C
- Einfache Ein- und Ausgabe mit *scanf* und *printf*

Aufgabe 3: „Aliquot Game“

Das Ziel dieser Aufgabe ist es, ein kleines Spiel zu programmieren, in diesem Fall das sogenannte Aliquot Game. Ein aliquoter Teiler (echter Teiler) einer natürlichen Zahl ist ein Teiler, der nicht identisch mit der Zahl selbst ist. So hat zum Beispiel die Zahl 12 die aliquoten Teiler 1, 2, 3, 4 und 6. Primzahlen haben die 1 als einzigen aliquoten Teiler. Die 1 selbst hat keinen aliquoten Teiler. Das „Aliquot Game“ ist ein Spiel für zwei Spieler, die abwechselnd ziehen. Das Spiel beginnt mit einer beliebigen natürlichen Zahl. Der Spieler, der am Zug ist, wählt einen der aliquoten Teiler aus und zieht diesen von der Zahl ab. Der Spieler, der die 1 gezeigt bekommt (und aus diesem Grund nicht mehr ziehen kann), verliert das Spiel.

Ein Spielablauf könnte so aussehen:

```
thales$ ./aliquot
*** Aliquot Game ***
Number: 65
Your move: 2
2 is not a proper divisor of 65!
Your move: 5
Number: 60
My move: 15
Number: 45
Your move: 15
Number: 30
My move: 15
```

```
Number: 15
Your move: 5
Number: 10
My move: 5
Number: 5
Your move: 1
Number: 4
My move: 1
Number: 3
Your move: 1
Number: 2
My move: 1
You lose!
thales$
```

Das Spiel soll so programmiert werden, dass ein menschlicher Spieler gegen den Computer antritt. Das Programm soll dem Spieler die aktuelle Zahl präsentieren und ihn dann nach einem aliquoten Teiler fragen, den es anschließend vom Terminal einliest. Es sollen nur valide Antworten zugelassen werden. Der Gegenspieler soll ein Teil des Programmes selbst sein. Im einfachsten Fall kann der Gegenspieler immer die 1 wählen. Ihm ein wenig mehr Intelligenz zu spendieren, wäre aber auch nicht verkehrt.

Hinweise:

Der Return-Wert von *scanf* und die Benutzereingaben sind zu überprüfen. Wenn keine Zahlen eingegeben werden, soll eine entsprechende Fehlermeldung ausgegeben und die Ausführung des Programms mit *exit(1)* abgebrochen werden (hierzu wird **#include** <stdlib.h> benötigt). Wenn die eingegebene Zahl kein aliquoter Teiler ist, soll das ebenfalls abgefangen und eine entsprechende Meldung ausgegeben werden. Beim Ende der Eingabe liefert *scanf* *EOF* zurück. Eine Zufallszahl lässt sich mit Hilfe der Bibliotheksfunktion *rand()* ermitteln, wobei der Pseudozufallszahlengenerator zuvor mit *srand()* in geeigneter Form initialisiert werden sollte, etwa mit Hilfe des Systemaufrufs *time()*, wofür die Header-Datei *time.h* einzubinden ist.

Einreichen können Sie Ihre Lösung mit folgendem Kommando auf der Thales:

```
submit ssl 3 aliquot.c
```

Wenn Sie Ihre Lösung einreichen, wird diese einer Reihe automatisierter Tests unterzogen, über die sie sofort interaktiv und auch per E-Mail einen Bericht erhalten. Bei den Tests gibt es sowohl solche, die nur den Quelltext betrachten, als auch solche, die Ihr Programm mit Testdaten ausführen. Es ist dabei durchaus möglich, dass die statischen Tests etwas beanstanden, das in Ordnung ist. Die Laufzeittests können auch mit Meldungen abbrechen wie „Execution aborted by signal“ mit einer Signalnummer, die größer oder gleich 30 ist. Dann

wurde es wegen Ressourcenverbrauch beendet. Das kann passieren, wenn Ihr Programm z.B. in eine Dauerschleife gerät. Bei diesem Übungsblatt ist das leicht möglich, wenn Sie das Resultat von *scanf* nicht korrekt überprüfen. Wenn Sie Fragen dazu haben, können Sie uns gerne kontaktieren.

Im übrigen werden die Lösungen verwendet, um ein generelles Feedback in der nächsten Übungsstunde zu geben.

Wenn Sie im Team arbeiten sollten, können Sie bei dem *submit*-Befehl auch noch eine *team*-Datei hinzufügen, bei der jede Zeile aus einem Loginnamen bei uns besteht. Alle Mitglieder müssen bei SLC für diese Vorlesung registriert sein. Den Testbericht mitsamt der eingereichten Lösung erhalten dann alle Teammitglieder per E-Mail.

Viel Erfolg!