

Blatt 02

Abgabe bis: 03. November 2011

Das “Aliquot Game”

Spielen macht Spass. Deshalb sollt ihr ein kleines Spiel programmieren. Als Mathematiker und Informatiker bevorzugen wir natuerlich mathematische Spiele. ;-) In diesem Fall das “Aliquot Game”.

Ein aliquoter Teiler (= echter Teiler) einer natuerlichen Zahl ist ein Teiler, der nicht identisch zur Zahl selbst ist. So hat beispielsweise die Zahl 12 die aliquoten Teiler 1, 2, 3, 4 und 6. Bei Primzahlen ist die 1 der einzige aliquote Teiler. Die 1 selbst hat keine aliquoten Teiler.

Das “Aliquote Game” ist ein Spiel fuer zwei Teilnehmer, die abwechselnd ziehen. Das Spiel beginnt mit einer beliebigen natuerlichen Zahl. Der Spieler, der am Zug ist, waehlt einen der aliquoten Teiler aus und zieht diesen von der Zahl ab. Der Spieler, der die 1 praesentiert bekommt (und dadurch nicht mehr ziehen kann, weil die 1 keine aliquoten Teiler hat) verliert das Spiel.

Quelle: A. Bogomolny, “Aliquot game” from Interactive Mathematics Miscellany and Puzzles; <http://www.cut-the-knot.org/SimpleGames/Aliquot.shtml>, Accessed 26 October 2011

Ein Spielablauf koennte so aussehen:

\$./aliquot	Bot: 7
*** Aliquot Game ***	Number: 7
Number: 59	You: 1
You: 2	Number: 6
No proper divisor!	Bot: 3
You: 70	Number: 3
Out of range!	You: 1
You: 1	Number: 2
Number: 58	Bot: 1
Bot: 29	You lose!
Number: 29	\$
You: 1	
Number: 28	
Bot: 7	
Number: 21	
You: 7	
Number: 14	

3. aliquot.c (8P)

Programmiert das Spiel so, dass ein menschlicher Spieler gegen den Computer antritt. Das Programm soll dem Spieler die aktuelle Zahl praesentieren und ihn dann nach einem aliquoten Teiler fragen, den es anschliessend vom Terminal einliest. Es soll nur valide Antworten zulassen. Der Gegenspieler soll Teil des Programms selbst sein. Im einfachsten Fall waehlt er immer die 1. Ein wenig mehr Intelligenz solltet ihr ihm aber schon spendieren.

Eine Zufallszahl laesst sich mit Hilfe der Bibliotheksfunktion `rand()` ermitteln. Sie gibt eine ganze Zahl zwischen 0 und `RAND_MAX` zurueck. Mittels der Modulo-Operation kann sie auf einen kleineren Bereich projeziert werden. Hinter `rand` steckt ein Pseudozufallszahlengenerator, der initialisiert werden sollte. Die Prozess-ID (`getpid()`) oder die aktuelle Zeit (`time()`) sind geeignete Startwerte. Eine Zahl im Intervall [10..59] bekommt man so:

```
srand(time());  
number = rand() % 50 + 10;
```

Informiert euch in den Manpages welche Headerdateien dazu eingebunden werden muessen. (`time` ist ein Systemaufruf und findet sich deshalb in Section 2.)

`printf` und `scanf` sind geeignete Funktionen zur Ein- und Ausgabe.

Zum Einreichen eurer Loesung:

```
submit ssl 3 team [notes] aliquot.c
```

4. Ganzzahlrepraesentation (2P)

Stellt die Zahlen 0, 42, 99, -1, -23, -128 binaer in 8bit dar. Verwendet das Zweierkomplement zur Darstellung der negativen Zahlen. Beispiel:

```
39 = 0010 0111  
-9 = 1111 0111
```

Zum Einreichen eurer Loesung:

```
submit ssl 4 team [notes] ganzzahlen.txt
```

Fragen zur Selbstkontrolle

Die folgenden Fragen muesst ihr nicht beantworten, ihr solltet sie aber beantworten koennen.

- Welche Arten von Pfadnamen gibt es bei Unix?
- Wie sieht die uebliche Form eines Shellbefehlsaufrufs (z.B. von `cp`) aus?
- Was sagt dir "a.out"?
- Nenne ein paar Kontrollkonstrukte von C?
- Wie sieht ein minimales C-Programm aus?
- Wie kompiliert man ein einfaches Programm?

- Nenne einen Unterschied zwischen Java und C.
- Wie gibt man Text auf der Standardfehlerausgabe aus?
- Wie kann man tabellarische Ausgaben erzeugen?
- In welcher Beziehung zueinander stehen die Ganzzahl-Datentypen in C?