

## Praktikumsversuch 1: Sieb des Eratosthenes

Angewandte Informatik · Wintersemester 2015/16 · Prof. Dr. Peter Gerwinski

Aufgabe: Schreiben Sie ein Programm, das das *Sieb des Eratosthenes* implementiert und veranschaulicht.

- Eine *Primzahl* ist eine positive Zahl, die durch genau 2 Zahlen teilbar ist, nämlich durch 1 und durch sich selbst.
- 1 ist *keine* Primzahl.  
(1 ist nur durch 1 Zahl teilbar, nämlich 1 selbst.)
- Die kleinsten Primzahlen lauten: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, ...
- Primzahlen sind diejenigen Zahlen, die „im Einmaleins übrig bleiben“: Zu den meisten Zahlen von 1 bis 100 findet man eine Aufgabe des kleinen Einmaleins, die diese Zahl als Lösung hat. Bei Primzahlen gibt es höchstens die „Aufgabe“ mit 1, z. B. „ $1 \cdot 7 = 7$ “.  
Auf dieser Eigenschaft beruht die als *Sieb des Eratosthenes* bezeichnete Methode, Primzahlen zu finden.
- Anwendung finden Primzahlen u. a. auf dem Gebiet der Kryptographie (Verschlüsselung, digitale Signatur).
- Für die Durchführung dieser Programmieraufgabe benötigen Sie *Arrays*. Diese funktionieren in C wie folgt:
  - Mit `int a[42]` deklarieren Sie eine Variable `a`, die 42 ganze Zahlen speichern kann.
  - Mit `a[i]` greifen Sie auf die *i*-te Komponente des Arrays zu.
  - Der Index beginnt grundsätzlich bei 0, geht in diesem Beispiel also von 0 bis 41.Mehr zu diesem Thema können Sie am Donnerstag in der Vorlesung erfahren.
- Machen Sie sich mit dem *modulo*-Operator `%` vertraut: `a % b` ist der Rest, der bei Division von `a` durch `b` zurückbleibt.  
Insbesondere ist `a % b` genau dann gleich 0, wenn `a` durch `b` teilbar ist.
- Machen Sie sich mit dem *Sieb des Eratosthenes* vertraut, z. B. per Web-Recherche.
- Der Wikipedia-Artikel zum *Sieb des Eratosthenes* illustriert den Algorithmus durch eine Animation.  
Schreiben Sie ein Programm, das eine vergleichbare Animation erzeugt, z. B. durch Ausgabe mit `printf()`.

Viel Erfolg!

Wer findet die größte Primzahl? ;–)