Aufgabe 1. Bringe GMP zum laufen. In cygwin sind die folgenden Packete Voraussetzung:

- gcc-4
- gmp
- libgmp-devel

Spiele ein wenig damit herum. Einige Fingerübungen:

a) Schreibe ein Programm, dass zwei große ganze Zahlen auf der Kommandozeile entgegen nimmt und ihr Produkt ausgibt.

b) Implementiere einen Primzahltest. Lerne danach, wie man den Primzahltest aus der GMP benutzt.

c) Schreibe eine Funktion, die mithilfe des euklidischen Algorithmus den ggT zweier großen Ganzzahlen berechnet.

d) Implementiere ein Modul, dass den Restklassenring modulo $m \in \mathbb{Z}$ implementiert.

Aufgabe 2. Implementiere den

Algorithmus 1 Miller–Rabin Test

Input: Zahlen $a, n \in \mathbb{N}$ mit $a < n$.
Output: "Ja", falls a Zeuge für n, andernfalls "Nein".

```
1: if n gerade oder ggT(a, n) ≠ 1 then
2:     return "Ja"
3: end if
4: Schreibe n − 1 = 2^k · q mit q ungerade.
5: set a := a^q mod n
6: if a ≡ 1 (mod n) then
7:     return "Nein"
8: end if
9: for i = 1 to k do
10:    if a ≡ −1 (mod n) then
11:        return "Nein"
12:    else
13:        set a := a^2 mod n
14:    end if
15: end for
16: return "Ja"
```

und verwende ihn, um eine Funktion zu schreiben, die zu einer gegebenen Bitlänge n eine Primzahl mit mindestens n Bits findet.