


1. Implementieren Sie den erweiterten euklidischen Algorithmus: Für ganze Zahlen a, b sind der $\text{ggT}(a, b)$ und ganze Zahlen x, y zu bestimmen, sodass $\text{ggT}(a, b) = xa + yb$. Begründen Sie die Korrektheit Ihres Verfahrens.

Hinweis: Starten Sie mit der Matrix $\begin{pmatrix} a & 1 & 0 \\ b & 0 & 1 \end{pmatrix}$ und führen Sie elementare Zeilen- und Spaltenoperationen durch, die durch Division mit absolut kleinstem Rest induziert werden.

2. Implementieren Sie den Chinesischen Restsatz für ganze Zahlen: Für gegebene $(a_1, \dots, a_r) \in \mathbb{Z}^r$ und paarweise teilerfremde $(m_1, \dots, m_r) \in \mathbb{Z}^r$ soll ein $x \in \mathbb{Z}$ bestimmt werden, sodass für alle $j \in \{1, \dots, r\}$ die Beziehung $x \equiv a_j \pmod{m_j}$ gilt.

3. Die Goldbach'sche Vermutung besagt, dass jede gerade Zahl > 2 Summe zweier Primzahlen ist. Schreiben Sie ein Programm, das die Vermutung für alle geraden Zahlen bis 10^5 überprüft.

4.  Seien eine Primzahl p und ein irreduzibles Polynom $f \in \mathbb{F}_p[X]$ vom Grad n gegeben. Implementieren Sie einen Datentyp für Elemente des Körpers $\mathbb{F}_{p^n} \simeq \mathbb{F}_p[X]/(f)$. Es soll zumindest die folgende Funktionalität implementiert sein:

- Darstellung als Polynom vom Grad kleiner als n
- Addition und Multiplikation
- Vergleich ($=$)
- Effizientes Potenzieren mit ganzen Zahlen (also auch Inversion)

Hinweis: In Sage ist dazu eine Klasse mit geeigneten Methoden `__add__`, `__mul__`, etc. zu definieren; in Mathematica sind die Operationen `Plus`, `Times`, etc. mittels `/:` zu definieren; in Maple ist das Beispiel ohne größeren Aufwand nicht lösbar. Effizientes Potenzieren z.B. mit `square-and-multiply`; Inversion entweder über Potenzieren oder mit erweitertem euklidischen Algorithmus.

5. Erklären Sie anhand von einfachen Beispielen die Funktionalität zum Rechnen mit endlichen Körpern in Mathematica, Maple und Sage.