Praktikum

Das Praktikum wird in 2-3 Teams bearbeitet und die **korrekte** Bearbeitung **aller** Aufgaben ist Voraussetzung für das Testat. Der Bericht, welcher alle Lösungen erfasst, wird in dem in moodle bereitgestellten Template bis zur dort angegebenen Abgabefrist bearbeitet und in moodle hochgeladen.

Gestalten Sie die Implementierung so, dass sie bei allen Aufgaben wiederverwendet werden kann. Die Aufgaben hängen von einander ab!

Für jede Aufgabe gilt: Erzeugen Sie Testfälle und Beispiele.

Für die Aufgaben werden keine Testvektoren zur Kontrolle zur Verfügung gestellt.

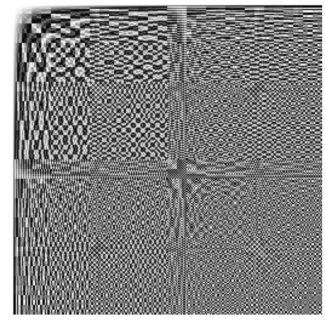
Tipp: Generieren Sie solche Testfälle selbst und tauschen Sie diese im Moodle-Forum mit den anderen Praktikumsteams aus.

Wer alle Algorithmen implementiert und auf Papier Testfälle und Beispiele durchgerechnet, hat den Großteil des Inhalts abgedeckt und ausreichend für die Klausur vorbereitet.

Aufgabe 1. Schreiben Sie ein Programm, welches die Multiplikationstabellen für Körper $\mathrm{GF}(2^e)$ konstruiert. Die Eingabe des Programms soll das irreduzible Polynom erwarten. Testen Sie das Programm für die Erweiterungsgrade $e = 2, 3, \ldots, 8$. Die entsprechenden irreduziblen Polynome über $\mathrm{GF}(2)$ finden Sie in der Tabelle am Ende des Kapitels über endliche Körper.

(Ein Test zur Korrektheit der Multiplikation wäre die Berechnung abc für Körperelemente $a, b, c \in \mathrm{GF}(2^e)$ auf zwei Arten $(ab)c$ und $a(bc)$. Sind beide Werte identisch ist das ein Indiz dafür, dass die Multiplikation korrekt arbeitet.)

(Die Multiplikationstabelle für $\mathrm{GF}(2^8)$ stellt sich wie am Rand abgebildet dar, wenn man die 256 Elemente 00000000, 00000001, ..., 11111111 mit Graustufen von weiß bis schwarz interpretiert.)



Aufgabe 2. Der erweiterte euklidische Algorithmus (EEA) für binäre Polynome bietet die Möglichkeit der multiplikativen Inversion in Erweiterungskörpern. Beschreiben und implementieren

Sie die multiplikative Inversion in binären Erweierungskörpern $\mathrm{GF}(2^e)$.

(Für verschiedene Körpergrade $e = 2, 3, \ldots, 8$ wählen Sie alle Körperelemente, invertieren diese mittels EEA und multiplizieren das Ergebnis mit dem Ausgangselement. Ist das Ergebnis 1, so war die Inversion korrekt.)

Aufgabe 3. Schreiben Sie ein Programm zur Behandlung von linearen Codes über einem binären Erweiterungskörper $\mathrm{GF}(2^e)$:

1. Implementieren Sie ein Routine, welche bei Eingabe einer Generatormatrix die kanonische Generatormatrix erzeugt.
2. Implementieren Sie eine Routine, welche aus einer kanonischen Generatormatrix eine Kontrollmatrix generiert.
3. Implementieren Sie eine Routine, welche die Syndromtabelle erzeugt.
4. Implementieren Sie eine Routine, welche die Fehlerkorrektur mittels Maximum-Likelihood-Decodierung und Syndromtabelle durchführt.

Aufgabe 4. Schreiben Sie ein Programm zur Behandlung von binären Hamming-Codes.

1. Implementieren Sie eine Routine, welche bei Eingabe von $m \geq 3$ eine Kontrollmatrix eines binären $(2^m - 1, 2^m - 1 - m, 3; 2)$ Hamming-Codes erzeugt (inkl. der dazugehörigen Generatormatrix).
2. Implementieren Sie die Decodierung für binäre $(2^m - 1, 2^m - 1 - m, 3; 2)$ Hamming-Codes.

Aufgabe 5. Schreiben Sie ein Programm zur rekursiven Konstruktion der Generatormatrix eines Reed-Muller-Codes $RM(r, m)$.

Aufgabe 6. Schreiben Sie eine Programm, welches bei Eingabe der Parameter e und $2 \leq d \leq 2^e - 2$ eines Reed-Solomon-Codes $RS(2^e, d)$ konstruiert. Konkret gehen Sie dabei wie folgt vor:

1. Implementieren Sie eine Routine zur Bestimmung eines primitives Element in $\mathrm{GF}(2^e)$.

2. Implementieren Sie eine Routine zur Berechnung des Generator- und Kontrollpolynoms für $RS(2^e, d)$.

3. Implementieren Sie eine Routine zur Berechnung von Generator- und Kontrollmatrix aus Generator- und Kontrollpolynom für $RS(2^e, d)$.

4. Implementieren Sie eine Routine zur Berechnung einer weiteren Kontrollmatrix mittels Vandermonde-Matrix.