Brief course on the mathematics behind coding theory and cryptography.
Software Activities.

1 Computing requirements

If you have other versions give them try first. If not, the following ones work correctly.

1. Install the Java Development Kit JDK version 8

   `https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html`

2. Install IDE Netbeans 8.1.

   `https://netbeans.org/downloads/old/8.1/`

2 Course Software Project

The Course Software is in the `studentSoftware` folder. With Netbeans browse inside this folder and open the `exercices` project.

2.1 Adding Documentation (Javadoc)

The Course Software is fully documented. Attaching the documentation to the software has to be done manually. The following link

`http://wiki.netbeans.org/FaqJavaDoc#Adding_Javadoc_via_the_Library_Manager`

provides information how to do so. The `javadoc` documentation is in the relative path:

```
/studentSoftware/exercices/dist/javadoc
```

To access the documentation, right click on an item in source file you are editing and click `Show Javadoc`.

Alternatively you can go to the

`/studentSoftware/exercices/dist/javadoc` folder and double click the `index.html` file. To get a full view, click the `FRAMES` tab.

3 Fields

Solve the following exercices by hand. Then open the course software project with Netbeans and implement the file `FieldsToComplete.java`

1. Implement the addition and multiplication tables of the field of 5 elements. Compute the inverses and check them using the previous multiplication table.

   `Use GaloisField gf5 = new RootGaloisField(5);`

2. Implement the addition and multiplication tables of the field of 4 elements. Compute the inverses and check them using the previous multiplication table.

   `Use GaloisField gf4 = new ExtendedGaloisField(gf2, 2);`

3. Check that $gcd(x^{15}-1, x^{20}-1) = x^5 - 1$.

   `Use GaloisField.Element[] coefsP1 = gf2.oneElement(),`

   `gf2.zeroElement(),gf2.oneElement();`

4. In $F_2[x]$, factor the following polynomials x^2+1, x^2+x+1, x^2+x.

   `Compare Vector<GaloisField.Element> rootsP5 = p5.roots();`

   `GFPolynomialFactorSet factorsP5 = p5.squareFree();`

5. In $F_{13}[x]$, let $a(x) = x^8 + x^6 + 10x^4 + 10x^3 + 8x^2 + 2x + 8$ and $b(x) = 3x^6 + 5x^4 + 9x^2 + 4x + 8$ Find two polynomials $q(x)$ and $r(x)$ such that

   $$a(x) = q(x)b(x) + r(x).$$

   `Use GaloisField gf13 = new RootGaloisField(13);`

   `GaloisField.Element[] elems13 = gf13.element;`

6. Given $p(x) = x^2 - x - 1$ in $F_3[x]$, construct the field $F_3[x] \mod p(x)$, by giving the addition and multiplication tables. Do the same with $p(x) = x^2 + 2$. What do you observe?

   `Use GaloisField.Element[] coefsX3 =`

   `gf3.zeroElement(),gf3.oneElement();`

   `GFPolynomial px3 = new GFPolynomial(coefsX3, gf3);`

7. Given $p(x) = x^2 - 2$ in $F_5[x]$, construct the field $F_5[x] \mod p(x)$, and compute the powers of $\alpha = (1,0)$. Do the same with $p(x) = x^2 + 2x + 3$. What do you observe?

8. Using F_{13} show that if t is the order or α, then t divides $q - 1 = 12$.

9. Using F_{16} show that there are $\phi(t)$ elements of order t, for appropriate t.

10. In F_7 find a primitive root and compute its powers.

11. in F_{25} find a primitive root and its primitive polynomial.

4 Error correction

Read Chapter 1, Sections 1.1, 1.2, 1.3, 1.4 of the excellent notes by Atri Rudra https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/chapters/chap1.pdf

4.1 Reed-Solomon codes

Read Chapter 5, Sections 5.1, 5.2, 5.3 of the excellent notes by Atri Rudra https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/chapters/chap5.pdf

4.2 Exercises

Start Netbeans and open the course project that you find in Atenea. Implement the following code:

1. `MainReedSolomonToComplete`

4.3 List Decoding Reed-Solomon codes

Read Chapter 13, Sections 13.1, 13.2 of the excellent notes by Atri Rudra The link says 12, but actually points to Chapter 13. https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/chapters/chap12.pdf

4.4 Exercises

Start Netbeans and open the course project that you find in Atenea. Implement the code in the following order:

1. `MyFactorizationToComplete`. You have A folder `factoring` inside the folder `notes`, that has a file `rothruckensteinstudents` with information on how to complete the factoring algorithm exercise.

2. Copy and paste your implementation in `MainReedSolomonToComplete` to test `MyFactorizationToComplete`.