19) Let $\beta \in \mathbb{F}_8$ be a root of $x^3+x+1 \in \mathbb{F}_2$. (In GAP you can thus chose $\beta = Z(8)$.) The polynomial

$$x^{20}+\beta^5x^{19}+\beta^2x^{18}+\beta^5x^{17}+\beta^2x^{16}+\beta^6x^{15}+\beta^2x^{14}+\beta x^{13}+\beta^6x^{12}+\beta^3x^{11}$$
$$+\beta^4x^{10}+\beta^4x^9+\beta^4x^8+\beta^6x^7+\beta^4x^6+\beta^6x^5+\beta^6x^4+\beta^6x^3+\beta^6x^2+x+\beta^4 \in \mathbb{F}_8[x]$$

is (over $\mathbb{F}_8$) a product of irreducible factors of degree 4. Determine these factors using the "trace" version of the Cantor-Zassenhaus method shown in the lecture.

20) Using Yun's algorithm, perform a squarefree factorization of the (multiplied out! The factor notation is only because it is shorter to type in) rational polynomial

$$(x-1)^3(x^2-3x-1)^4(x^2-2x-4)(x^2-x+4)(x^2-3)^3(x^2+x+1)^2(4x^3-x^2-x-1)^3$$

21) Factorize the (squarefree part of) the polynomial from problem 16 using Berlekamp's algorithm.

22) For a natural number n define the **Möbius function** as:

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1 \\ (-1)^k & \text{if } n \text{ is the product of } k \text{ distinct primes} \\ 0 & \text{if } n \text{ is not squarefree} \end{cases}$$

a) Show that if $m,n \in \mathbb{N}$ are coprime, then $\mu(mn) = \mu(m)\mu(n)$.
b) Show that $\sum_{d|n} \mu(d) = 0$ if $n > 1$, where the sum is over all positive divisors of n.
c) Let R be a ring and $f,g\colon \mathbb{N}_{>0} \to R$ two functions with

$$f(n) = \sum_{d|n} g(d).$$

Show that

$$g(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) f(d) = \sum_{d|n} \mu(d) f\left(\frac{n}{d}\right) \quad \text{for } n \in \mathbb{N}_{>0}.$$

d) Now assume that $f(n) = \prod_{d|n} g(d)$ for $n \in \mathbb{N}_{>0}$. Give a similar formula for g in terms of μ and f as in c) e) Give a formula for the number of irreducible polynomials of degree d over the field with q elements.
f) Determine the number of conjugacy classes of GL(3,5).
f') (non-M567): Describe an algorithm that determines all irreducible polynomials of degree d over the field of q elements which uses as little polynomial factorization as possible.

Saving your work in GAP

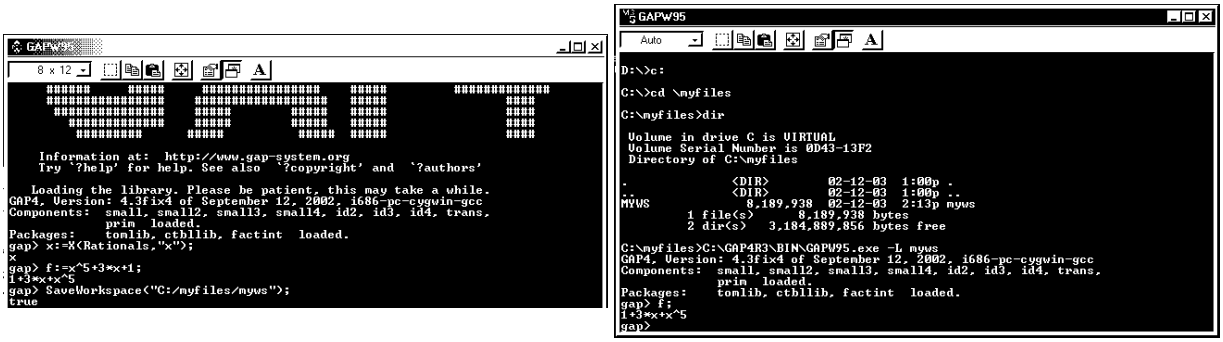
It is possible (unless you are in a `brk`-loop) to save the current state of a GAP session. This is done with the command `SaveWorkspace(filename);` where filename is the file in which the workspace will be stored. (You must have permission to write it, thus under Windows you likely will have to give a full path.)
To restore a saved GAP session, you start GAP with the command line option -L and give the saved workspace as parameter. Again this is obvious under Unix, under Windows it might require some contortion:

1. Open the DOS-shell (MS-DOS icon).
2. go in the disk and the directory that contains the saved workspace:

   ```
   C:
   cd \myfiles
   ```

3. start the GAP binary (with its full path) and give the workspace as parameter for the `-L` option:

   ```
   C:\GAP4R3\BIN\GAPW95 -L myws
   ```



More remarks about polynomials in GAP

As GAP can create multivariate polynomials, it is possible to create different indeterminates over the same field, these are internally labelled by their number (a positive integer).
If `f` is a polynomial (or an indeterminate) the command
`IndeterminateNumberOfUnivariateLaurentPolynomial` (this might be a good point to mention that the <TAB> key can be used to complete a partially typed command...) returns its indeterminate number.
The number `1` as last parameter in the `UnivariatePolynomial` command as given on a previous sheet is exactly such an index number, one could use 2,3 or other and create a polynomial in a different indeterminate.
Caveat: The command `X(Rationals,"x")` takes a (new) indeterminate number, returns the corresponding indeterminate, and sets printing for it to `x` (one could call it differently). The string given as name has no implicit correlation to the index number – anything is possible, in particular creating several indeterminates that display with the same name.
Caveat 2: If you issue several commands `X(Rationals,"x")` in one GAP session, you create several different indeterminates, which have different index number but all display as `x`. In this case, you will have to replace the 1 in an `UnivariatePolynomial` call by the correct index number, you can find this out with `IndeterminateNumberOfUnivariateLaurentPolynomial(x)`. (Otherwise you may get results that look weird as different variables are named the same.)
Caveat 3: Indeterminate numbers are independent for each characteristic. It might be that you work with indeterminate number 2 in characteristic zero and indeterminate number 1 in characteristic 3.