Problem Sheet 1 (Jul. 30)

1) Solve the equation $5x \bmod 123 = 7$.

2) In $GF(49)$, let α be a root of -1. Calculate $1/\alpha + \alpha$.
GAP **Commands**: `X`, `RootsOfUPol`

3) Explain the output of the following commands. How could you evaluate f at 1?

```
gap> x:=X(Rationals,"x");
x
gap> f:=x^2+x+1;
x^2+x+1
gap> x:=1;
1
gap> f;
x^2+x+1
```

4) Create a list of the first 20 cubes.
GAP **Commands**: `List`, `[1..20]`

5) Create – with minimal GAP-assignments – the list $[1, 2, 8, 7, 17, 5, 4, 3, 9, 10]$.

6) Determine all Elements $x \in \mathrm{GF}(32)$ such that $x^5 + x^2 + 1 = 0$.
GAP **Commands**: `Filtered`, `GF`, `Z(2)^0`

7) Determine all generators of $\mathrm{GF}(27)^*$.

8) A number n is called *perfect* if the sum of its divisors (denoted by $\sigma(n)$) equals $\sigma(n) = 2n$. Find all perfect numbers up to 10^6.
GAP **Commands**: `Filtered`, `Sigma`

9) The command `Combinations([1..5])` returns all subsets of the set $\{1, \ldots, 5\}$.
a) Find all subsets of $\{1, \ldots, 15\}$, whose entries add up to 15.
GAP **Commands**: `Filtered`, `Sum`, `Combinations`
b) Repeat the problem with subsets of $\{1, \ldots, 22\}$ which sum up to 22. What happens? Can you modify the command to avoid the problem?
GAP **Commands**: `Combinations([1..22],i)` for $1 \leq i \leq 22$, `List`, `Concatenation`.

10) Let $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$. Determine all solutions to the system of equations $x \cdot A = (10, 10, 10)$ and $A \cdot y = (-10, -11, -12)^T$.
GAP **Commands**: `SolutionMat`, `NullspaceMat`, `TransposedMat`

10a) A string in GAP is simply a list of characters and can be manipulated accordingly. Thus for example

```
s:="TUCSON";
Arrangements(s,3);
```

returns all 3-letter sequences that can be made from the letters in the word "TUCSON".
a) Read in the file `words.g`. This file defines a variable `words`, which is a list of strings for English words. Using this list, find all 3-letter combinations of "TUCSON", which are proper English words.
b) We now want to find all words that can be made from the letters of "CANTALOUPE", but already the 4-letter combinations take rather long to test. (You can use the command `time;` after a command to find out how many milliseconds it took.) Now issue the commands

```
words:=List(words,Immutable);;
IsSSortedList(words); # tests that the list is sorted.
```

and repeat the commands. Why is it suddenly so much faster?
c) (Once you know a bit more about groups in GAP) The number of arrangements of a given length (you can calculate this by `NrArrangements(s,k);`, e.g.) goes up substantially. Instead use `Combinations` and then run over all permutations in S_n using `Enumerator(SymmetricGroup(k))` (which does not write down all elements and takes little memory). Using this, determine how many enlish words can be formed from the letters of "TOBEORNOTTOBETHATISTHEQUESTION".

11) Let

```
m:=GeneratorsOfGroup(Centre(GL(2,5)))[1];
```

Create a new matrix, in which the last row is replaced by its negative.
GAP **Commands**: `ShallowCopy`

12) Write a function that adds the odd-position entries of a list. Apply it to `[1..100]`, `[5,3,7]` and `[]`.

13) We define the Tribonacci numbers by $T_0 = 1$, $T_1 = 3$ and $T_n = T_{n-1} + T_{n-2}$ for $n > 1$. a) Write a function to calculate these numbers and use it to calculate T_{10000}.
b) Determine the last 2 digits of $T_{10^{10^{10}}}$.
GAP **Commands**: `PowerMod`

14) The following function was written to create all vectors of length 3 with entries in $\{0,1\}$. But what does it create? Why? Can you fix the code?

```
l:=[];
# create vectors recursively, by trying out all entries at position 'pos':
allVectors:=function(v,pos)
local i;
  for i in [0..1] do
    v[pos]:=i;
    if pos=3 then
      Add(l,v);
    else
      allVectors(v,pos+1);
    fi;
  od;
end;

allVectors([0,0,0],1);
l;
```