© Judy Holdener, July 27, 2001

Number of Groups of a Given Order

As the title indicates, this lab will explore the number of possible group structures for any given order.

Instructions: Refer to the attached **gap** session printout when working through the questions that follow. Most of the calculations you do will be similar to those in the printout.

Question 1. Before answering this question, use **gap** to produce a list of pairs [i, N_i], where i is an integer between 1 and 100 and N_i is the number of (solvable) groups. Amazingly, there is exactly one group (A_5) of order less than 100 which is not solvable, and that won't be an obstruction to answering the questions in this lab.

Using the output, classify those integers that lead to a larger number of group structures. That is, for which orders are there numerous groups (up to isomorphism)? For which orders are there few groups (up to isomorphism)?

Question 2. Are the primes the only orders for which there is only one group (up to isomorphism)?

Question 3. Is there any pattern to the orders for which there is only one group (up to isomorphism)?

Question 4. To answer this question you will want to produce the following:

- a list of the number of groups of order $2p$, where p is prime and $2p \leq 100$
- a list of the number of groups of order $3p$, where p is prime with $p > 3$ and $3p \leq 100$.

Then produce a few other lists similar to those above but corresponding to different primes. If you understand **gap**'s list function well then you can probably figure how to create a list whose elements are the lists for each prime.

Using the information you just produced, refine your answer to question 3. Is there a pattern for orders that are a product of 2 distinct primes? Note that 2 behaves a bit differently, but it's not really an exception.

Question 5. State and prove a conjecture concerning the number of groups (up to isomorphism) of order pq where p and q are distinct primes.

© Judy Holdener, July 27, 2001

Appendix.

```
gap> # list of the first ten squares
gap> List([1..10],i->i^2);
[ 1, 4, 9, 16, 25, 36, 49, 64, 81, 100 ]
gap> # list of all groups of orders 1 through 12. The answer is a list of lists.
gap> List([1..12],i->AllSolvableGroups(Size,i));
[ [  ], [ c2 ], [ c3 ], [ 2^2, c4 ], [ c5 ], [ c6, S3 ], [ c7 ],
  [ 2^3, 4x2, c8, D8, Q8 ], [ 3x3, c9 ], [ c10, D10 ], [ c11 ],
  [ 6x2, c12, D12, 6.2, A4 ] ]
gap> # To get the number of groups of each order, ask for the length of the list
gap> # of those groups.
gap> List([1..12],i->Length(AllSolvableGroups(Size,i)));
[ 0, 1, 1, 2, 1, 2, 1, 5, 2, 2, 1, 5 ]
gap> # Let's add the order.  We get a list of pairs: [i, number of groups of size i]
gap> List([1..12],i->[i,Length(AllSolvableGroups(Size,i))]);
[ [ 1, 0 ], [ 2, 1 ], [ 3, 1 ], [ 4, 2 ], [ 5, 1 ], [ 6, 2 ], [ 7, 1 ],
  [ 8, 5 ], [ 9, 2 ], [ 10, 2 ], [ 11, 1 ], [ 12, 5 ] ]
gap> # the Filtered command 'filters' a list: for example, we can compute the
primes:
gap> List([1..100],i->IsPrime(i));
[ false, true, true, false, true, false, true, false, false, false, true,
  false, true, false, false, false, true, false, true, false, false, false,
  true, false, false, false, false, false, true, false, true, false, false,
  false, false, false, true, false, false, false, true, false, true, false,
  false, false, true, false, false, false, false, false, true, false, false,
  false, false, false, true, false, true, false, false, false, false, false,
  true, false, false, false, true, false, true, false, false, false, false,
  false, true, false, false, false, true, false, false, false, false, false,
  true, false, false, false, false, false, false, false, true, false, false,
  false ]
gap> Filtered([1..100],i->IsPrime(i));
[ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
  73, 79, 83, 89, 97 ]
gap> # Find all non-abelian groups of order 24:
gap> Filtered(AllSolvableGroups(Size,24),x->not IsAbelian(x));
[ D8x3, Q8x3, S3x2^2, S3x4, 2x6.2, 12.2, A4x2, grp_24_11, D24, Q8+S3,
  Sl(2,3), S4 ]
gap> # primes p such that 3p<600:  Here we use the pre-defined list of primes less
gap> # than 1000  Note that we get the primes, not the 3p's.
gap> Filtered(Primes,p->3*p<600);
[ 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71,
  73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151,
  157, 163, 167, 173, 179, 181, 191, 193, 197, 199 ]
```