Math 6833 assignments

3. Get started on GAP:

   (a) Try to download GAP and get it running on your machine.

   (b) Once you have GAP installed, the next task is to be able to read the manual. The best way is to put a link on your browser which takes you to the html version of the manual that is on you own computer after you install GAP. In case you have difficulty, I have put links to various versions of the manual on our links page.

   (c) Start playing with the tutorial. Probably you will want to start in section 2.

4. Find a GAP exercise and work on it:

   1. The best exercise is something you want to try doing in your research, or for another course you may be taking, or just because you are interested in it.
   2. If nothing occurs to you, here is a project: Starting with the cFrac.gap code (available on the links page), do more with continued fractions. For example, every rational number can be written as a continued fraction $[2a_1, b_1, 2a_2, b_2, \ldots, 2a_n]$ or $[2a_1, b_1, 2a_2, b_2, \ldots, 2a_n, b_n]$, if we allow some of the a_i and b_i to have different signs. Write code to perform expansions of this type. Some rational numbers, but not all, can be expanded in the form $[2a_1, 2a_2, \ldots, 2a_n]$. Write code to perform this expansion. Do some experiments and formulate a conjecture about which continued fractions can be expanded in this form. Prove it mathematically.

5. Here is another possible GAP exercise.

   1. Write a routine to take an $m \times n$ matrix, regarded as a presentation of an abelian group, and print out the abelian group it presents (the GAP routine SmithNormalFormIntegerMat will save you a lot of time, as well as the pretty printing routine printAbelianGroup.gap on our links page).
   2. Extend your abelian presentation finder to a group presentation abelianizer, that is, given a presentation, you print out its abelianzation. For example, you might try to set up a presentation such as $\langle x, y, z \mid x^2y^2 = y^2x^2, z^6 \rangle$ using lists like this:
      ```
      generators := [ "x", "y", "z" ];
      relation1 := [ [ "x", 2 ], [ "y", 2 ], [ "x", -2 ], [ "y", -2 ] ]
      relation2 := [ [ "z", 6 ] ]
      relationList := [ relation1, relation2 ]
      ```
      (You will need to read about how to use characters and strings in GAP.)
   3. Read about how to use group presentations in GAP, and figure out how to use its built-in functions to find group abelianizations. You can use these to check the results of your own code.
   4. If you want, figure out how to do this using Magnus.