Mid-Semester Project

Math 242

due Wednesday, April 11

Use Mathematica to investigate some mathematical ideas. Almost anything is fair game: find one or more questions, statements, or ideas to explore. The process is more important than the topic! You do not need to prove any theorems.

The following list contains some ideas and links to papers that may provide inspiration for your project. However, don't be limited by this list — feel free to come up with your own idea!

- Finite Sums of the Alcuin Numbers
- Conway's Subprime Fibonacci Sequences
- Double Fun with Double Factorials
- Prime Number Races
- Use computation to solve a Putnam problem
- Proof Without Words: Limit of a Recursive Root Mean Square
- A Spigot Algorithm for the Digits of Pi
- PSLQ: An Algorithm to Discover Integer Relations
- On an Intriguing Integral and Some Series Related to $\zeta(4)$

Group work is allowed. The professor wants to know your intended topic and group by Friday, April 6.

As usual, submit code that runs and explain what your code does; make it clear that you know how your implementation works. Your goal should be to communicate your work to another person (e.g., another student at your level who is not in this course).

Your notebook will be graded on a scale of 0 to 4, according to the following rubric.

4. Problems and goals are clearly stated, including relevant definitions or parameters. Computations are complete; code runs and is clearly explained. Conclusions are clearly stated and backed up by sufficient computational evidence. Limitations of the methodology, extensions for future work, and/or conjectures are discussed. Notebook is well-formatted and easy to read.

3. Problems and goals are stated well, though relevant definitions or parameters may be missing. Computations are mostly complete; code runs, but explanation is weak. Conclusions are unclear or not well justified. Insufficient discussion of limitations, extensions, and/or conjectures.

2. Statement of problem or goal is unclear. Computations are incomplete; explanation is ambiguous. Code may produce errors when run. Conclusions are possibly correct, but not justified. Little or no discussion of limitations, extensions, and/or conjectures. Notebook is difficult to read.

1. Serious misunderstanding of the problem or goal. Computation is inadequate for the task at hand. Work is not clearly explained. No discussion of limitations, extensions, and/or conjectures. Notebook is difficult to read.

0. Notebook is not turned in.