All students should read §2.1–2.2 of the textbook in preparation for the next lecture. Skip the first part of the §2.2 *Implementation* section, start reading again at the *Randomness* subsection marker.

After reading from the text, read a tale of `casino theives` . It is a non-technical (public press) description of how some pRNGs have been exploited in the past. Be prepared to connect the concepts described in this article to the nuts and bolts of pRNGs in the text.

The following numbered questions should be split across your group and the solutions discussed during the next lecture period. Students should review the learning goals for the day, determine which are applicable to their questions and provide answers or commentary to their group members. When using the Internet to formulate answers (some questions may require this), keep track of **where** you find your information on the web. You may be asked for, and are expected to have (in Email-able form), URLs supporting your investigations.

1. **Warning! Heavy Duty Reading!** Read `errors in Monte Carlo sims` and provide a written summary to your group. It should be at least a paragraph long.

   This is a **dense** but short paper. You should look up the meaning of any random number terms or names you find. But in general you can avoid investigating details of the actual **system** they are simulating — but you should describe to your group the types of problems and erroneous results the group found having to do with types of pRNGs used and that "random numbers fall mainly on the plains" (or *lattice* in this case).

2. (a) Answer question 2.1.1 (§2.1.3) — it should go without saying but I'm expecting people to write **code** for this, not perform the math by hand!

   (b) Investigate the effect of hardware limited arithmetic on this generator. In this case $m = 127 = 2^7 - 1$, suppose your computer's machine word is limited to the 7 bits required to hold 127. Each arithmetic operation would then be $\bmod 128$ (just like in a 32 bit machine all integer results are $\bmod 2^{32}$).

   Make a second version of your program that mimics the results from this type of hardware.

   i. How many "full period multipliers" exist for this **faulty** generator?

   ii. Are any full period multipliers shared with the proper generator of part a?

3. Write code for the generator presented in question 2.1.1 (§2.1.3) and find at least one full period multiplier. Reproduce a graph such as Figure 2.2.2 of the text showing that "random numbers fall mainly on the planes." For bragging rights, show a similar figure in 3 dimensions for the $(a,m) = (66,401)$ generator presented in the book. Choose a viewing angle for your graph that makes it easy to see the "lattice" of planes.

4. Time for a history lesson! Read `good pRNGs are hard to find`; some of it is skimmable material that we have already covered in lecture; the discussion of portable implementations will be of interest to (I suspect) a minority of students (so don't worry too much about it), but certainly read the A SAMPLING OF INADEQUATE GENERATORS section (beware the toxic levels of academic disdain) and provide a written summary of **new** information to your group. It should be at least a paragraph long.