Homework 5

This homework is due Thursday, Mar. 1 (by 5:00 PM).

1. (20 points) At the height of the structured programming controversy, L. Flon published a short paper ("On Research in Structured Programming", *ACM SIGPLAN Notices*, Vol. 10, Num. 10, (Oct. 1976), pages 16-17) where he advanced the following "axiom:"

   > There is not now, nor will there ever, exist a programming language in which it is the least bit hard to write bad programs.

   His point was that every feature of a programming language could be misused, so endless debates about the harmful effects of features (such as the `goto`) were silly.

   In this problem you are going to test Flon's axiom for yourself. Your task is to write the worst Pascal program that computes the greatest common divisor of two positive integers (see below). The program must meet the following criteria:

   - The program must read two positive integers from the Unix standard input, and then print their greatest common divisor (just the number), followed by a newline. No other I/O (e.g., prompting) is allowed. The greatest common divisor of two integers is the largest integer that evenly divides both. The greatest common divisor of 0 and 0 is defined to be 0. (The standard input is obtained by reading from the Pascal text file `input`.) Please test your program before turning it in.

     Thus your program should not be bad just because it does not work, it should just be badly written.
   - The program must be less than 5000 characters long (including comments). No line of the program may extend past the 80th column when displayed with tab positions at every 8 tab stops. Use the program `wc` to count characters.
   - The program must not use the `goto` statement. (That would be too obvious.)
   - The program, when run, should compute its answer in less than a minute on Zippy on legal Pascal integers. (So your algorithm shouldn't be *too* awful.)

   We will exhibit the most humorous program meeting these criteria in class.

   Be creative: your points for this problem will partly depend on how "bad" your program is (within the above guidelines). (Hint: it may be best to make the program "bad" at a "high level"; a badly formatted program is not very funny. You might try first writing a "good" program and then introduce bad points.)

   Turn in a printout of your program and also mail the source for your program to `mobasher` on Zippy. Be sure to include your name at the beginning of the file in a comment.

2. (5 points) Briefly state why your program for the first problem is bad. (Hint: what principles of good design does it violate?)

3. (10 points) If one can write bad programs in any language, what makes a language good?