HW04 CMSC/MATH/ENEE 456. Morally DUE Oct 12

1. (0 points)
   (a) What is the day and time of the midterm?
   (b) IF you CANNOT make the timed part of the midterm let me know NOW!!!!!!!!!!

GOTO NEXT PAGE

2. (25 points, programming question) The goal of this problem is to (1) get data on what fraction of numbers are safe primes, (2) write programs that will be used for both Diffie-Helman and RSA.

   (a) (0 points) Program EXP. On input a, n, p, output $a^n \pmod p$. Make it efficient, so use repeated squaring. Some languages have this built in, but you are not allowed to use it.

   (b) (0 points) Program TESTPRIME. Test for primality using the following method which is a variant of what was on the slides: To test if n is prime pick 5 distinct random numbers $a_1, a_2, a_3, a_4, a_5 \in \{2, \ldots, n-2\}$ and compute, for $1 \leq i \leq 5$, $a_i^{n-1} \pmod n$. If ALL are 1 then output 1. if ANY are not 1 then output 0. (So 1 means PRIME and 0 means NOT PRIME.)

   For $n \leq 7$, you will not be able to pick 5 distinct numbers, so you can hard-code the result for all n such that $1 \leq n \leq 7$.

   (c) (0 points) Program TESTSAFEPRIME. Given a number n, test if its a SAFE prime. If it is then output 1, if not then output 0.

   (d) (25 points) Program HOWMANYSAFEPRIME: Given n, determine how many numbers in $\{1, \ldots, n\}$ are safe primes.

   In your main method, you should take as input n and output the resulting integer from HOWMANYSAFEPRIME(n).

   (a) n will be given as a command line argument. Expect your filename to be the first command line argument and n to be the second. There will be no input given through standard input.

   (b) You should output HOWMANYSAFEPRIME(n) to standard output.

   (c) You should upload a **single** file ending in `.java`, `.py`, `.ml`, `.rb`, `.c`, `.cpp`, or `.scala`, corresponding to Java, Python3, OCaml, Ruby, C, C++, and Scala respectively.

GOTO NEXT PAGE

3. (25 points, written question) You will use your programs from question 2 for the following:

   (a) (15 points) Run HOWMANYSAFEPRIME on the inputs 10000, 20000, ..., 90000. Use this to determine what proportion of numbers in $\{1, ..., 10000\}, \{1, ..., 20000\}, ..., \{1, ..., 90000\}$ are safe primes.

   Report your results.

   (b) (10 points) Based on this data make a conjecture about

$$f(x + 10000) - f(x)$$

   where f is HOWMANYSAFEPRIME.

GOTO NEXT PAGE

4. (25 points, programming question) The goal of this problem is to (1) get data on what fraction of numbers are generators, (2) write programs that will be used for both Diffie-Helman and RSA.

   (a) (0 points) Program TESTGEN. Given p and g do the following

      i. Test if p is a safe prime (if NOT then output 2 and stop, so 2 means BAD INPUT because NOT a safe prime.)

      ii. Test if $g \in \{2, \ldots, p-2\}$ (if NOT then output 3 and stop, so 3 means BAD INPUT because g is not in the right range).

      iii. (If you got this far then p is a safe prime and g is a candidate for a generator.) Find $q = \frac{p-1}{2}$. Note that this will be a prime. Compute $g^2 \pmod{p}$ and $g^q \pmod{p}$. If BOTH are not 1 then g is a generator. If EITHER is 1 then g is not a generator. Output 1 if g is a generator and output 0 if g is not.

   (b) (25 points) Program HOWMANYGEN: Given p (test if p is a safe prime and if its not output "*not safe man!*") determine how many numbers in $\{2, \ldots, p-1\}$ are generators.

   In your main method, you should take as input p and output the result from HOWMANYGEN(p).

   (a) p will be given as a command line argument. Expect your filename to be the first command line argument and p to be the second. There will be no input given through standard input.

   (b) You should print HOWMANYGEN(p) to standard output, which should be either an integer or "*not safe man!*"

   (c) You should upload a **single** file ending in `.java`, `.py`, `.ml`, `.rb`, `.c`, `.cpp`, or `.scala`, corresponding to Java, Python3, OCaml, Ruby, C, C++, and Scala respectively.

GOTO NEXT PAGE

5. (25 points, written question) You will use your programs from question 4 for the following:

   (a) (25 points) Run HOWMANYGEN on:
   1019, 2027, 3023, 4007, 5087, 6047, 7079, 8039, 8963, and 10007
   Use this to determine the following

      i. What proportion of numbers in $\{2, ..., 1019-1\}$ are generators of 1019?
      ii. What proportion of numbers in $\{2, ..., 2027-1\}$ are generators of 2027?
      iii. What proportion of numbers in $\{2, ..., 3023-1\}$ are generators of 3023?
      iv. DOT DOT DOT
      v. What proportion of numbers in $\{2, ..., 10007-1\}$ are generators of 10007?

   Report your results.

   (b) (0 points) Based on this data make a conjecture about $g(p)$, where g calculates the proportion of generators in $\{2, ..., p-1\}$ of p.