CMSC 456 Untimed Midterm, Fall 2021

1. This is an open-book, open-slides, open-web exam. This is also a week long
2. There are 3 problems which add up to 50 points.
3. In order to be eligible for as much partial credit as possible, show all of your work for each problem, **write legibly**, and **clearly indicate** your answers. Credit **cannot** be given for illegible answers.
4. After the last page there is paper for scratch work.
5. Please write out the following statement: "*I pledge on my honor that I will not give or receive any unauthorized assistance on this examination.*"

6. Fill in the following:

NAME :
SIGNATURE :
UID :

THERE ARE THREE PROBLEMS
MAKE SURE YOU DO ALL OF THEM

1. (20 points) In class Dr. Gasarch made the following statement:

   If you take a long enough text T (length $|T|$) and then look at every xth letter, the letter frequencies will be similar to that of normal English.

   This is true so long as $x \ll |T|$. In this problem we will see how big x has to be before it stops being true.

   You may use *your own* code from previous homeworks for this question.

   (a) (0 points) Write a program that does the following:

      i. Input: a text of English T (intent: a normal text) and a number $x < |T|$.

      ii. Output: a 26-long freq vector $f_{T,x}$ of every xth letter of the text. (Remember: the freq vector is a vector of probabilities).

      iii. Advice: Run L2N on T and then do the count.

      When counting every xth letter, index starting at 0 and include every letter if its index is divisible by x.

      For example, if we processed T ="`What's the weather like today?`" where $x = 4$, we should get "`wswhio`". You would proceed to obtain $f_{T,4}$ from this new processed text.

   (b) (20 points) Write a program that does the following:

      i. Input: a text of English T (intent: a normal text).

      ii. What the Program Does: For $1 \leq x \leq |T|$ find the DOT product of $f_{T,x}$ and the freq of English. Call this DOT product $d_{T,x}$.
      Make sure you calculate $|T|$ with all special characters, numbers, spaces, etc. removed.

      iii. Output a table of the $d_{T,x}$. Describe what you see. In particular, how large does x have to be before every-xth-letter-of-T no longer is close to English.

   GOTO NEXT PAGE FOR HOW TO SUBMIT

(a) Expect raw input data to potentially be a large text with special characters, both uppercase and lowercase letters, new-lines, tabs, spaces, etc. Output text should have the output with only the letters remaining (case-insensitive).

(b) Inputs are everything read through standard input (stdin) and outputs should be printed to standard output (stdout). Your program should only be able to process one text at a time - multiple lines are still treated as part of the same text.

(c) Your program should output $|T|$ lines, with each line containing only $d_{T,x}$ **starting** from $x = 1$ and **ending** at $x = |T|$.

(d) Do NOT round your floats. Print them with as many digits as you have, and the autograder will round them internally.

(e) The ENG vector will be provided to you in a separate file called `engvector.csv`.

(f) 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. If applicable, use the default package.

GOTO THE NEXT PAGE FOR NEXT PROBLEM

2. (20 points) Find a 3×3 matrix M that maps FBI to CIA. NOTE: The matrix NEED NOT be invertible!!!! (We use A is 0, B is 1, etc.)

GOTO THE NEXT PAGE FOR NEXT PROBLEM

3. (10 points) The goal of this problem is to code up Diffie Helman. You may use *your own* code from previous homeworks for this question.

   (a) FINDSAFEPRIME. On input L output a safe prime that is between 2^L and $2^{L+1}-1$ (so its L bits long) by doing the following: pick a random number r between 2^L and $2^{L+1}-1$ and test if its a safe prime. If so GREAT. If not then try $r+1, r+2, \ldots$ until you get one. (IF it ends up being over $2^{L+1}-1$, thats fine.)

   (b) FINDGEN. Given p a safe prime (if its not a safe prime output an appropriate insult) find a generator for p by testing random numbers in $\{2, ..., p-1\}$ until you get one. (NOTE- in the real world you would not do it this way, but on the HW we do it this way so its easier for you to do and for us to grade.)

   (c) SETUPDH. On input L, output a safe prime p and a generator for it g.

   (d) DHAlicesends. On input (p, g) pick a random $a \in \{2, \ldots, p-2\}$ and output $g^a \pmod p$.

   (e) DHBobsends. On input (p, g) pick a random $b \in \{2, \ldots, p-2\}$ and output $g^b \pmod p$.

   (f) DHAlicegetskey. On input (p, a, x) compute $x^a \pmod p$. Note that if $x = g^b \pmod p$ then this will be $g^{ab} \pmod p$.

   (g) DHBobgetskey. On input (p, b, x) computes $x^b \pmod p$. Note that if $x = g^a \pmod p$ then this will be $g^{ab} \pmod p$.

GOTO NEXT PAGE FOR SUBMISSION DETAILS

In your main method, you should take as input L and output many different values computed throughout this problem.

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

(b) You should output many different variables to standard output on separate lines as follows (exponentiation is done (mod p)):

i. On the first line, print your safe prime, p, from SETUPDH.

ii. On the second line, print your generator, g, from SETUPDH.

iii. On the third line, print your random value for a from DHAliceSends.

iv. On the fourth line, print your computed value for g^a from DHAliceSends.

v. On the fifth line, print your random value for b from DHBobSends.

vi. On the sixth line, print your computed value for g^b from DHBobSends.

vii. On the seventh line, print your computed g^{ab} from DHAliceGetsKey.

viii. On the eighth line, print your computed g^{ab} from DHBobGetsKey. Note: this should match what is printed on the previous line. As confirmation that your programs work properly, it is highly recommended that you compute this value independently.

(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.