Fall 2024 CIS 3362 Homework #5: Number Theory
Check WebCourses for the due date

1) (5 pts) Without the aid of a computer program, determine the prime factorization of 16,141,806,000. Show your work. You may do division on a calculator. Stating which numbers divided in evenly how many times.

2) (5 pts) What is φ(16,141,806,000)? You can use a calculator, but please show your work.

3) (5 pts) Use Fermat's Theorem to calculate the remainder when 5^{6879} is divided by 983?

4) (5 pts) Use Euler's Theorem to calculate the remainder when 38^{104835} is divided by 10829?

5) (10 pts) Show the steps of running the Miller-Rabin algorithm, testing n = 1705 for primality with the randomly chosen value of a = 3. Please use a calculator or computer program to calculate the modular exponents and just show the result of each squaring/mod operations

6) (10 pts) Trace through the Fermat Factoring algorithm to factor 245,239 as the product of two prime numbers. You may use a calculator or computer program to execute each calculation, but print out the result of each number being tested as a perfect square.

7) (5 pts) A primitive root, α, of a prime, p, is a value such that when you calculate the remainders of α, α^2, α^3, α^4 , ... , α^{p-1}, when divided by p, each number from the set {1, 2, 3, ..., p-1} shows up exactly once. Prove that a prime p has exactly φ(p-1) primitive roots. In writing your proof, you may assume that at least one primitive root of p exists. (Normally, this is the first part of the proof.) (Note: This question is difficult, so don't feel bad if you can't figure it out.) **(Note: The solution to this can probably be found on the internet, so I'll be looking for original explanations that show understanding but aren't identical to the book proofs…ie what a normal person would come up with after thinking about the problem on their own)**

8) (5 pts) In class, we made a chart, for p = 7, of the different lengths of cycles produced by exponentiating each of the possible non-zero mod values, mod 7. We found that two of the values (3, 5) have a cycle length of 6, two of the values (2, 4) have a cycle length of 3, 1 value (6) has a cycle length of 2, and 1 value (1) has a cycle length of 1. Based on this example, give a counting/logical argument proving the sum below, for prime numbers, p:

$$\sum_{d \in Divisor(p-1)} \phi\left(\frac{p-1}{d}\right) = p - 1$$

Again, this is something you can find online, but it's just a really cool problem to think about and I wanted to give those of you who are interested in thinking about something like this an opportunity to do so. Don't feel bad if you can't do it. To be fair to students who would like to challenge themselves, please don't look up the proof idea. If I detect a decent amount of copying for this one, I'll throw it out of the grading criteria. Same for #7.

9) (50 pts) In order to determine if a number g is a primitive root of a prime, p, we must simply take each divisor, d, of p – 1, and calculate the remainder when $g^{\frac{p-1}{d}}$ is divided by p. If none of these remainders equals 1, then g is a primitive root of p. Write a program that reads in input from standard input using the format described below, and for each pair of input values, p and g, respectively, determines if g is a primitive root of the prime p. You may assume that p is a prime number. For each input case, you'll output 1 if g is a primitive root mod p and 0 if it's not a primitive root. **(Half credit will be given for correct solutions that have a run time of O(p), where p is the prime number in the input. In order to get full credit, you must use an algorithm that takes roughly O($\sqrt{p}$) time to find the divisors of p – 1, followed by utilizing fast modular exponentiation.)**

Please write your program in Java, Python, C, or C++ and submit your code file only, naming it primrootcheck.ext, where ext = "java" or"py" or "c", or "cpp".

Input Format

The first line of input will contain a single positive integer, $\boldsymbol{n}$ ($1 \leq \boldsymbol{n} \leq 50$), indicating the number of input cases.

Each subsequent line will contain one input case each. This will consist of the integer, $\boldsymbol{p}$ ($11 \leq \boldsymbol{p} < 10^{12}$) followed by the integer $\boldsymbol{g}$ ($1 < \boldsymbol{g} < \boldsymbol{p-1}$), where $\boldsymbol{p}$ is prime.

Output Format

For each case, on a line by itself, output "1" (without the quotes) if $\boldsymbol{g}$ is a primitive root of $\boldsymbol{p}$. Otherwise, output "0" on a line by itself if $\boldsymbol{g}$ is not a primitive root of $\boldsymbol{p}$.

Note: A scaffold for your program has been provided for you in C on the next page. Please carefully look at it and use it (or translate to another language.)

Program Scaffold

Here is a scaffold for the program in C, so that all of you do the input/output correctly. Translate accordingly to other languages as needed.

```
#include <stdlib.h>
#include <stdio.h>

// Put all function prototypes here.
int isPrimRoot(long long p, long long g);

int main() {

    int n;
    scanf("%d", &n);

    for (int loop=0; loop<n; loop++) {

        long long p, g;
        scanf("%lld%lld", &p, &g);

        if (isPrimRoot(p, g))
            printf("1\n");
        else
            printf("0\n");

    }

    return 0;
}

int isPrimRoot(long long p, long long g) {

    // Put code here.

    // Just to make this compile.
    return 1;
}

/*** Other functions go here ***/
```