Fall 2024 CIS 3362 Homework #5: Number Theory Solutions

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.

Solution

16,141,806,000 / 1000 = 16,141,806 Divide by 1000 to get a smaller number

1000 / (2^3) = 125 / (5^3) = 1 Find the prime factorization of 1000

16,141,806 / (2^1) = 8,070,903 Divide by 2 until no longer divisible

8,070,903 / (3^2) = 896,767 Divide by 3 until no longer divisible

Continue to do this for each prime number, skipping 5 because we already removed it

Can't divide 7, 11, or 13.

896,767 / (17^2) = 3103

Can't divide 19 or 23

3103 / (29^1) = 107

107 is prime.

Bring it all together,

16,141,806,000 = (2^4) * (3^2) * (5^3) * (17^2) * (29^1) * (107^1)

Check using a calculator to make sure that your prime factorization is correct.

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

Solution

φ(16,141,806,000) = φ(2^4) * φ(3^2) * φ(5^3) * φ(17^2) * φ(29^1) * φ(107^1)
φ(16,141,806,000) = ($2^4 - 2^3$) ($3^2 - 3^1$) ($5^3 - 5^2$) ($17^2 - 17^1$) ($29^1 - 29^0$) ($107^1 - 107^0$)
φ(16,141,806,000) = 8 * 6 * 100 * 272 * 28 * 106
φ(16,141,806,000) = 3,875,020,800

In prime factorized form, its: 2^3 * (2^1 * 3^1) * (2^2 * 5^2) * (2^4 * 17^1) * (2^2 * 7) * (2^1 * 53^1)
= 2^{13} * 3^1 * 5^2 * 7^1 * 17^1 * 53^1
(PS I had to check my work 3 times on this one to make sure I did it right.)

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

Solution

Since 983 is prime and GCD(5, 983) = 1, by Fermat's Theorem, we have $5^{982} = 1 \bmod 983$.

$5^{6879} = 5^{7(982)+5} = (5^{982})^7 * 5^5 \equiv 1^7 * 3125 \equiv 3125 \bmod 983 \equiv 176 \bmod 983$

The desired remainder is 176

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

Solution

$10829 = 7^2 * 221 = 7^2 * 13 * 17$

It follows that $\varphi(10829) = \varphi(7^2) * \varphi(17) * \varphi(13) = (7^2 - 7)(17 - 17^0)(13 - 13^0) = 42 * 16 * 12 = 8064$

Since the GCD(38, 10829) = 1, due to Eulers Theorem, it follows that $38^{8064} \equiv 1 \bmod 10829$.

So:

$38^{104835} = 38^{13(8064)+3} = (38^{8064})^{13} * 38^3 \equiv (1)^{13} * 54872 \bmod 10829 \equiv 727 \bmod 10829$

The remainder of 38^{104835} divided by 10829 is 727.

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

Solution

Using the Miller-Rabin test, we rewrite $1704 = 2^3 \times 213$

We first calculate $3^{213} \bmod 1705 = 368$, which leads us to square this value:

$368^2 \bmod 1705 = 729$, and then we'll square again:

$729^2 \bmod 1705 = 1186$

Since none of those three values were -1 mod 1705, we can conclude that 1705 isn't prime.

Note: In fact, for this case, $3^{1704} \not\equiv 1 \pmod{1705}$, so the basic Fermat Theorem test would have sufficed in this particular case.

Therefore, **1705 is composite.**

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.

$\sqrt{245,239} \sim 495.2$, thus, we can start our algorithm with x = 496

x	$X^2 - 245,239$	Is perfect square?
496	777	No
497	1770	No
498	2765	No
499	3762	No
500	4761	Yes (69 x 69)

It follows that 245,239 = (500 + 69)(500 – 69) = **569 * 431**

Using Fermat Factorization, we split 245,239 into two prime factors closest to its square root, and these are 569 and 431

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)**

Solution

Let α be an arbitrarily chosen primitive root of p.
Let k be an integer in between 1 and p – 1 such that gcd(k, p – 1) = 1.
Now consider the value $\beta \equiv \alpha^k (mod\ p)$. Our goal will be to prove that β is a primitive root as well.

Consider the sequence of values $\beta, \beta^2, \beta^3, \dots, \beta^{p-1}$ mod p. These will be equivalent to

$$\alpha^k, \alpha^{2k}, \alpha^{3k}, \dots, \alpha^{(p-1)k}$$

mod p, respectively.

We know the last of these is equivalent to 1 (mod p) via Fermat's Theorem.

What we would like to show is that these exponents, when taken mod (p – 1) will all be distinct values covering the set {0, 1, 2, …, p – 2}. If we can show that, then we know that the actual list of values itself are distinct mod p and that the first time the value of 1 shows up on the list is at the very end.

Assume to the contrary, that two values on the list of exponents, k, 2k, 3k, …, (p – 1)k are equivalent mod (p – 1). Then there exist distinct integers, i and j, with $1 \leq i < j \leq p - 1$, such that ki and kj are equivalent mod (p – 1):

$$kj \equiv ki \ (mod\ p-1)$$
$$kj - ki \equiv 0 \ (mod\ p-1)$$
$$k(j-i) \equiv 0 \ (mod\ p-1)$$

Because gcd(k, p – 1) = 1, it follows that $(j-i) \equiv 0 \ (mod\ p-1)$. (This uses a rule that was stated in class but not proved. Intuitively, if there are no common factors between k and p – 1, then for p – 1 to divide this product, it has to entirely divide into the other part and not k.)

But, recall that 0 < j – i < p – 1. This means that p – 1 can NOT divide the difference between j and i, resulting in a contradiction. It follows that the original assumption was incorrect, and that each of the exponents to alpha are distinct mod p – 1.

Since α is a primitive root, by definition, this list of values is equivalent to each unique non-zero value mod p. Thus, if α is a primitive root, it follows that β is as well. Thus, the number of primitive roots is at least the number of values k such that gcd(k, p – 1) = 1. By definition of the phi function, this is $\phi(p-1)$.

We must finally also prove that if gcd(k, p – 1) ≠ 1, then α^k is NOT a primitive root. Once we prove this, then we know the count above is accurate and not below the actual answer. (This is the only if part of the proof.) Let d = gcd(k, p – 1) > 1. Also, let $X = \frac{k}{d}$. We know that *X* must be an integer by definition of gcd.

Let $\beta = \alpha^k$. Consider the exponent $\frac{p-1}{d}$. Since g > 1, this exponent is strictly less than p – 1. Now, calculate the following:

$$\beta^{\frac{p-1}{d}} = \alpha^{\frac{k(p-1)}{d}} = \alpha^{X(p-1)} \equiv 1 \ (mod\ p)$$

This proves that the order of β is less than p – 1. Thus, β is not a primitive root of p.

Thus, we've proved that given one primitive root, α, all other primitive roots must be of the form α^k, where gcd(k, p – 1) = 1. By definition of the phi function, there are precisely $\phi(p-1)$ of these.

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:

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

Solution

The cycle length of each possible base, by Fermat's theorem, **MUST BE** a divisor of p – 1. Thus, if we were to make a frequency chart of how many bases have each cycle length (as described in the problem statement), the sum of those frequencies must necessarily equal p – 1, since there are p – 1 bases, 1 through p – 1, to consider.

Thus, what remains to be proven is that for any divisor, d, of p – 1, the number of elements with order $\frac{p-1}{d}$ is exactly $\phi(\frac{p-1}{d})$. If we can prove this, then symbolically, the sum on the left will represent the number of elements/bases of each different possible cycle length, and since each element must appear exactly once in the sum, it would then follow that the sum equals p – 1.

In question 7, we proved the fact specifically for the divisor d = 1.

Now, let's generalize that proof for any divisor d.
Consider a primitive root, α, we can generate each possible base mod p by exponentiating it to each power from 1 to p – 1. Every one of these values can be represented as $\beta = \alpha^k$, for some integer k in the range 1 to p – 1. Let d = gcd(k, p – 1).

As previously stated in the proof for #7, $\beta^{\frac{p-1}{d}} = \alpha^{\frac{k(p-1)}{d}} = \alpha^{X(p-1)} \equiv 1 \ (mod\ p)$.

From this statement, it follows that $\beta^{\frac{p-1}{d}(i)} \equiv 1 \ (mod\ p)$, for each integer in between 1 and d, inclusive. Thus, each of these values (there are d of them), have order, $\frac{p-1}{d}$, because the list of values, $\beta^1, \beta^2, \dots, \beta^{\frac{p-1}{d}}$ are all unique mod p, with the last value equivalent to 1 mod p. (To fully prove this, we can do another proof by contradiction rewriting each of these in terms of α, and proving that the list of exponents is unique mod (p – 1).

This proof works for each divisor, d, of p – 1. It follows that there are $\phi(\frac{p-1}{d})$ bases with an order of $\frac{p-1}{d}$. As previously discussed, this means that the sum on the left adds exactly 1 for each unique base modulo p. Thus, the sum must equal exactly p – 1.

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, ***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, ***p*** ($11 \leq \boldsymbol{p} < 10^{12}$) followed by the integer ***g*** ($1 < \boldsymbol{g} < \boldsymbol{p}$-***1***), where ***p*** is prime.

Output Format

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

Solution

Please see attached files primrootcheck.c, primrootcheck.py,
primroot1.in, primroot1.out (for both),
primroot2.in, primroot2.out (for Python only) and
primroot3.in, primtroot3.out (for C only).