**Question 1**

Your goal this week is to write a program to compute discrete log modulo a prime *p*. Let *g* be some element in  and suppose you are given *h* in  such that *h*=*gx* where 1≤*x*≤240. Your goal is to find *x*. More precisely, the input to your program is *p*,*g*,*h* and the output is *x*.   
  
The trivial algorithm for this problem is to try all 240 possible values of *x* until the correct one is found, that is until we find an *x* satisfying *h* = *gx* in Z*p*. This requires 240 multiplications. In this project you will implement an algorithm that runs in time roughly   = 220 using a meet in the middle attack.

Let *B*=220.Since *x* is less than *B*2 we can write the unknown *x* base *B* as where *x*0,*x*1 are in the range [0,*B*−1]. Then

By moving the term *gx*1 to the other side we obtain  in Z*p*.  
  
The variables in this equation are *x*0,*x*1 and everything else is known: you are given *g*,*h* and *B*=220. Since the variables *x*0 and *x*1 are now on different sides of the equation we can find a solution using meet in the middle ([Lecture 3.3](https://class.coursera.org/crypto-010/lecture/view?lecture_id=14)):

* First build a hash table of all possible values of the left hand side  for *x*1=0,1,…,220.
* Then for each value *x*0=0,1,2,…,220 check if the right hand side is in this hash table. If so, then you have found a solution (*x*0,*x*1) from which you can compute the required *x* as *x*=*x*0*B*+*x*1.

The overall work is about 220 multiplications to build the table and another 220 lookups in this table.   
  
Now that we have an algorithm, here is the problem to solve:

*p*=13407807929942597099574024998205846127479365820592393377723561443721764030073546976801874298166903427690031858186486050853753882811946569946433649006084171

g=11717829880366207009516117596335367088558084999998952205599979459063929499736583746670572176471460312928594829675428279466566527115212748467589894601965568  
  
*h*=3239475104050450443565264378728065788649097520952449527834792452971981976143292558073856937958553180532878928001494706097394108577585732452307673444020333

Each of these three numbers is about 153 digits. Find *x* such that *h*=*gx* in Z*p*.   
  
To solve this assignment it is best to use an environment that supports multi-precision and modular arithmetic. In Python you could use the [gmpy2](http://readthedocs.org/docs/gmpy2/en/latest/mpz.html#mpz-methods) or [numbthy](http://www.math.umbc.edu/~campbell/Computers/Python/numbthy.py) modules. Both can be used for modular inversion and exponentiation. In C you can use [GMP](http://gmplib.org/). In Java use a BigInteger class which can perform mod, modPow and modInverse operations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Your Answer** |  | **Score** | **Explanation** |
| 375374217830 | Correct | 1.00 |  |
| Total |  | 1.00 / 1.00 |  |