Cryptography Assignment - 4
Pramod Aravind Byakod

1: Compute the subgroup generated by 2 + 17 Z in (Z/17Z)^*

subgroup generated = {1,15,13,9}

2: Determine the order of all the elements in (Z/15Z)^*

Elements	Order	
1	1	1^1 = 1 mod 15
2	4	2^4 = 1 mod 15
4	2	4^2 = 1 mod 15
7	4	7^4 = 1 mod 15
8	4	8^4 = 1 mod 15
11	2	11^2 = 1 mod 15
13	4	13^4 = 1 mod 15
14	2	14^2 = 1 mod 15

3. In Sage, after initiation:
sage: R = Integers(23875916459823645643826545648564 87)
sage: a = 20973482746524897458296458 4
sage: b = 83457489574823658264875247 5485
if we run
sage: R(a)^b
we get the answer:
23416702453836441953378308613521 66
However, if we run
sage: R(a^b)
we get "RuntimeError". Explain why by estimating how much disk space (in GBytes) is needed to store the result of a^b in binary.

The exact error message is "`RuntimeError: exponent must be at most 9223372036854775807`".

This error is clearly because of the resultant number is big enough to not fit in the memory space allocated. Let's have a look at how much memory it would need for a RAM to store this resultant number.

Let's assume b = `9223372036854775807`

```
sage: b = 92233720368547758O7
sage: sys.getsizeof(int(R(a^b)))
python(23941,0x7fff8d8ca340) malloc: *** mach_vm_map(size=2305843009213698048) failed (error code=3)
*** error: can't allocate region
*** set a breakpoint in malloc_error_break to debug
---------------------------------------------------------------------------
MemoryError                               Traceback (most recent call last)
<ipython-input-129-3151318659fd> in <module>()
----> 1 sys.getsizeof(int(R(a**b)))

/Applications/SageMath-8.1.app/Contents/Resources/sage/src/sage/rings/integer.pyx in sage.rings.integer.Integer.__pow__ (build/cythonized/sage/rings/integer.c:14260)()
   2065         cdef Integer x = PY_NEW(Integer)
   2066
-> 2067         sig_on()
   2068         mpz_pow_ui(x.value, (<Integer>self).value, nn if nn > 0 else -nn)
   2069         sig_off()

MemoryError: failed to allocate 2305843009213693968 bytes
sage:
```

To store value of R(a^b), system is going to need 2305843009213693968 bytes, which is equivalent to 2305843009.21 GB.

In our case setting value of b to 834574895748236582648752475485 will require way more than 2305843009.21 GB for the system to store the resultant value of R(a^b) in binary. Which is impractical.

4. Search the Internet for information about the RSA challenge numbers. Prove that RSA-1024 is a composite number using the Fermat Little Theorem with "a" = your id number.

RSA-1024 is
135066410865995223349603216278805969938881475605667027524485143851526510604859
533833940287150571909441798207282164471551373680419703964191743046496589274256
239341020864383202110372958725762358509643110564073501508187510676594629205563
685529475213500852879416377328533906109750544334999811150056977236890927563

RSA-1024 has 1,024 bits (309 decimal digits) and has not been factored so far.
RSA numbers are a set of large semi primes (numbers with exactly two prime factors) that are part of the RSA Factoring Challenge. The challenge was to find the prime factors. It was created by RSA Laboratories to encourage research into computational number theory and the practical difficulty of factoring large integers.

Let's use contradiction to prove RSA 1024 is a composite number.
According to Fermat Little Theorem, If gcd(a, m) =1, then a^phi(m) = 1 mod m.
Consider m = RSA-1024 and a = 113436879
GCD(m,113436879) = 1

```
sage: gcd(113436879,135066410865995223349603216278805969938881475605667027524485143851526510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563)
....: 25762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563)
1
sage:
```

We got to prove, 113436879^phi(m-1) != 1 mod m

Since (m-1) is an even number, any number multiplied by an even number is again an even number. So, 113436879^phi(m-1) is an even number.

When we divide 113436879^phi(m-1) by m, we do not get 1 as a reminder. According to Fermat's theorem, our assumption contradicts, so RSA-1024 or m should not be a prime number, hence it's a composite number.