Description of the assignment.

The RSA modulus N is the product of two large primes, p and q. The vulnerability that will be exploited is that p and q are very close together, specifically satisfying:

$$|p - q| < 2N^{1/4}$$

We use Fermat's factorization method in this case because the two prime factors p and q are very close to each other, and that's exactly the situation where Fermat's method is most effective.

Assume: Since p and q are close, the arithmetic mean A = (p+q)/2 is close to $\sqrt{N}$. So we approximate: A ≈ $\lceil\sqrt{N}\rceil$.

Fermat's method leverages the identity: $N = p * q = (A - x)(A + x) = A^2 - x^2$

Compute $A = \lceil\sqrt{N}\rceil$

Computer $x^2 = A^2 - N$

Try to compute $x = \sqrt{x^2}$

Now, we can recover p and q: $p = A - x : q = A + x$

Finally, I verified by using p*q = N

When p and q are too close together, the value of x becomes small, so A^2 is very close to N. Thus, it's easy to compute $x = \sqrt{A^2 - N}$, and from there get p and q.

Screenshot of the output:

```
ryptography/assi/coding-4/Rsa.py
p = 134078079299425970995740249982058461274793658205923933777235614437217640300736627688911
11614362326998675040546094339320838419523375986027530441562135724301
q = 134078079299425970995740249982058461274793658205923933777235614437217640300737785609803
48930557750569660049234002192590823085163940025485114449475265364281
x = 57896044618658097711785492504343953926634992332820282019728792003956564819990
verify the N = p*q True
```

Description of the assignment.