1. (15 points) Let $a, b \in \mathbb{Z}$ with $a \geq b > 0$, and let $q = \lfloor a/b \rfloor$. Show that $\ell(a) - \ell(b) - 1 \leq \ell(q) \leq \ell(a) - \ell(b) + 1$, where $\ell(x)$ is the length of the binary representation of an integer x.

since $a \geq b > 0$ $\quad q = \lfloor a/b \rfloor \geq 1$

$\ell(q) = \lfloor \log_2 q \rfloor + 1$

$\ell(a) - \ell(b) = \lfloor \log_2 a \rfloor - \lfloor \log_2 b \rfloor \geq \lfloor \log_2 q \rfloor$

so $\lfloor \log_2 q \rfloor + 1 \leq \lfloor \log_2 a \rfloor - \lfloor \log_2 b \rfloor + 1$

that is $\ell(q) \leq \ell(a) - \ell(b) + 1$.

since $q = \lfloor a/b \rfloor$

then $\lfloor \log_2 a \rfloor - \lfloor \log_2 b \rfloor \leq \lfloor \log_2 q \rfloor + 2 = \lfloor \log_2 4q \rfloor$

that is $\ell(a) - \ell(b) \leq \ell(q) + 1$.

so $\ell(a) - \ell(b) - 1 \leq \ell(q)$

conclusion: $\ell(a) - \ell(b) - 1 \leq \ell(q) \leq \ell(a) - \ell(b) + 1$

```python
def ext_euclid(a, b):
    old_s, s = 1, 0
    old_t, t = 0, 1
    old_r, r = a, b
    if b == 0:
        return 1, 0, a
    else:
        while(r!=0):
            q = old_r // r
            old_r, r = r, old_r-q*r
            old_s, s = s, old_s-q*s
            old_t, t = t, old_t-q*t
    return old_s, old_t, old_r

a = int(input("输入第一个数字:"))
b = int(input("输入第二个数字:"))
s, t, r = ext_euclid(a, b)
print("s = %d, t = %d, r = %d" % (s, t, r))
print("%d*%d+%d*%d=%d" % (a, s, b, t, s*a+t*b))
```

```
#include <bits/stdc++.h>
using namespace std;
const int mod=1e9+7;
long long quick_mod(long long a,long long b)
{
    long long ans=1;
    while(b){
        if(b&1){
            ans=(ans*a)%mod;
            b--;
        }
        b/=2;
        a=a*a%mod;
    }
    return ans;
}
long long quickmod(long long a,char *b,int len)
{
    long long ans=1;
    while(len>0){
        if(b[len-1]!='0'){
            int s=b[len-1]-'0';
            ans=ans*quick_mod(a,s)%mod;
        }
        a=quick_mod(a,10)%mod;
        len--;
    }
    return ans;
}
 int main(){
    char s[100050];
    int a;
    while(scanf("%d",&a))
    {
        scanf("%s",s);
        int len=strlen(s);
        printf("%I64d\n",quickmod(a,s,len));
    }
    return 0;
}
```

4. (20 points) Solve the following linear congruence equations:

(1) $17x \equiv 11(\bmod 23)$;

(2) $55x \equiv 35(\bmod 75)$.

(1) $17x \equiv 34 \pmod{23}$

$x \equiv 2 \pmod{23}$

(2) $11x \equiv 7 \pmod{15}$

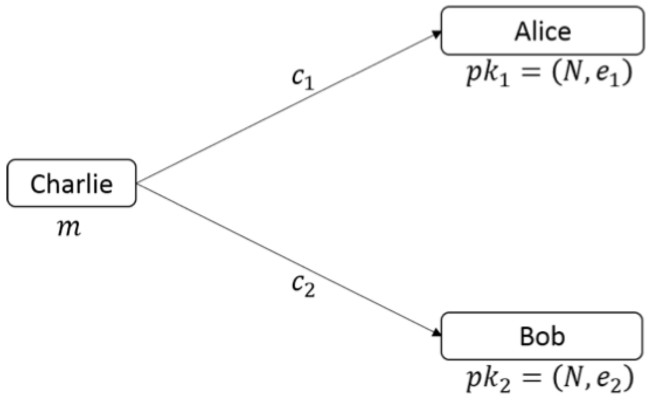
$11x \equiv 22 \pmod{15}$

$x \equiv 2 \pmod{15}$

So the solution is $x \equiv 2 + 15k \pmod{75}$

$(k = 0, 1, 2, 3, 4)$

5. (15 points) See the following figure. Alice and Bob trust each other very much. They set their RSA public keys as $pk_1 = (N, e_1)$ and $pk_2 = (N, e_2)$, respectively. Charlie wants to send a private message m to Alice and Bob, where $0 \le m < N$ is an integer and $\gcd(m, N) = 1$. To this end, Charlie encrypts m as $c_1 = m^{e_1} \bmod N$ and $c_2 = m^{e_2} \bmod N$; and then sends c_1 to Alice and sends c_2 to Bob.



Suppose that $\gcd(e_1, e_2) = 1$ and Eve sees all public keys and ciphertexts. Determine if Eve can learn the value of m.

$d_1 = \frac{k_1 \varphi(n) + 1}{e_1}$ $d_2 = \frac{k_2 \varphi(n) + 1}{e_2}$

$c^{d_1} \bmod N = m$ $c^{d_2} \bmod N = m.$

$m = \frac{(c^{d_1} + c^{d_2}) \bmod N}{2}$