整除

性质

$$
\begin{gathered}
1.\text{如果}a|b\text{且}b|c\text{，那么}a|c\text{。}\\
2.a|b\text{且}a|c\text{等价于任意的整数}x\text{和}y,\text{有}a|(b*x+c*y)\\
3.\text{设}m\neq 0,\text{那么}a|b\text{等价于}(m*a)|(m*b)\\
4.\text{设整数}x\text{和}y\text{满足下式：}a*b+b*y=1,\text{且}a|n\text{、}b|n\text{，那么}(a*b)|n
\end{gathered}
$$

奇妙的性质

1.如果2能整除a的末一位，则2|a

2.如果4能整除a的末两位，则4|a

3.如果8能整除a的末三位，则8|a

4.如果3能整除a的各位数字之和，则3|a

5.如果11能整除a的偶数位数字之和与奇数位数字之和的差，则11|a

6.如果一个数的末三位与末三位前面的数字组成的数之差能被7、11、13整除，那么这个数就能够被7、11、13整除。

二元一次不定方程

一般形式$ax+by=c$，此方程有整数解的充要条件是GCD(a,b)|c

$$
\begin{gathered}
\text{设}x_0,y_0\text{是该方程的一组整数解，那么该方程的所有整数解可表示为：}\\
x=x_0+\frac{b}{GCD(a,b)}t,y=y_0-\frac{a}{GCD(a,b)}t
\end{gathered}
$$

同余

性质

自反性：$a\equiv a(\mod m)$

对称性：若$a\equiv b(\mod m)$，则$b\equiv a(\mod m)$

传递性：若$a\equiv b(\mod m),b\equiv c(\mod m)$，则$a\equiv c(\mod m)$

同加性：若$a\equiv b(\mod m)$,则$a+c\equiv b+c(\mod m)$

同乘性：若$a\equiv b(\mod m)$，则$a*c\equiv b*c(\mod m)$，若$a\equiv b(\mod m)$，$c\equiv d(\mod m)$,则$a*c\equiv b*d(\mod m)$

同幂性：若$a\equiv b(\mod m)$，则$a^n\equiv b^n(\mod m)$

推论1：$a*b\mod k=(a\mod k)*(b\mod k)\mod k$

推论2：$$
\begin{gathered}
\text{若}a\mod p=x,a\mod q=x,p,q\text{互质，则一定存在则整数}s,t\text{，使得}a=s*p+x,a=t*q+x,\\
\text{所以},s*p=t*q,\text{则一定存在整数}r\text{，使}s=r*q,\\
\text{所以，}a=r*p*q+x\text{，得出}a\mod p*q=x
\end{gathered}
$$

最大公约数

二进制算法（非递归求GCD）

```
inline int gcd(int x,int y){
    int i,j;
    if(x==0) return y;
    if(y==0) return x;
    for(i=0;(x&1)==0;i++) x>>=1; //去掉所有的2
    for(j=0;(y&1)==0;j++) y>>=1; //去掉所有的2
    if(j<i) i=j;
    while(1){
        if(x<y) x^=y^=x^=y;//若x<y，交换x,y
        if((x-=y)==0) return y<<i;
        //若x==y，gcd==x==y（就是在辗转减，while(1)控制）
        while((x&1)==0) x>>=1; //去掉所有的2
    }
}
```

扩展欧几里得

```
inline int exgcd(int a,int b,int &x,int &y){
    if(!b){
        x=1;y=0;
        return a;
    }
    int d=exgcd(b,a%b,y,x);
    y-=a/b*x;
    return d;
}
```

求解线性同余方程

```
//用扩展欧几里得算法解线性方程：ax+by=c
bool linearEquation(int a,int b,int c,int &x,int &y){
    int d=ex_gcd(a,b,x,y);
    if(c%d) return false; //如果gcd(a,b)|c才有解
    int k=c/d;
    x*=k; //+ t*b;
    y*=k; //-t*a;
    //求的只是其中一个解
    return true;
}
```

快速幂

```
int fpow(int a,int p,int mod){ //计算a的p次方在模mod下的值
    int res=1;
    while(p){ //将a^p分治成 a^(p/2)*a^(p/2)，当p为奇数时乘上a
        if(p&1) res=res*a%mod;
        a=a*a%mod;
        p>>=1;
    }
    return res;
}
```

逆元

$a*b\equiv 1 \pmod b), a, b$互质，则称$x$为$a$的逆元，记为$a^{-1}$

求逆元的四种方法

①拓展欧几里得

②快速幂+费马小定理

③递推

```
inv[1]=1;
for(ull i=2;i<=n;i++){
    inv[i]=(ull)(p-p/i)*inv[p%i]%p;
}
```

快速乘

龟速乘

事实上快速乘是为了防止溢出，又不想写高精度，所以我们模仿二进制加法来完成两数的取模乘积。复杂度$O(logn)$

```
ll mul(ll x,ll y,ll mod){ //=>x*y%mod
    ll res=0;
    while(y){
        if(y&1) res=(res+x)%mod;
        x=(x+x)%mod;
        y>>=1;
    }
    return res;
}//其中ll是long long类型
```

优秀的long double快速乘

先上代码，复杂度只有$O(1)$噢

```
ll mul(ll a,ll b,ll mod){
    ull c=(ull)a*b-(ull)((ld)a/mod*b+0.5L)*mod;
} //其中ull是unsigned long long，ld是long double类型
```

仔细一看，这直接用了乘法操作不就爆掉了吗?

其实a*b和(a*b/p)*p两部分都是会溢出的，但是unsigned保证了他们溢出后的差值不变，因此不会影响最终结果。（反正很巧妙）

中国剩余定理(CRT)

"三人同行七十稀，五树梅花廿一枝，七子团圆月正半，除百零五便得知。"

首先引入同余方程

$$\begin{cases} x \equiv 2 \mod 3 \\ x \equiv 3 \mod 5 \\ x \equiv 2 \mod 7 \end{cases}$$

小模数分别为5*7=35，3*7=21，3*5=15，找乘法逆元

$$\begin{cases} 35a \equiv 1 \mod 3 \\ 21b \equiv 1 \mod 5 \\ 15c \equiv 1 \mod 7 \end{cases}$$

我们可以解得逆元a=2,b=1,c=1，于是有下列式子

$$\begin{cases} x \equiv 70 \mod 105 \\ x \equiv 21 \mod 105 \\ x \equiv 15 \mod 105 \end{cases}$$

这里70=2*5*7，21=3*7，15=3*5（都是不同方程里面的素因子）

我们可以求得$2*70+3*21+2*15 \mod 105 = 23$为x的解

定义

$$\text{设}m_1, m_2, \ldots, m_r\text{两两互素，并记}N = m_1 * m_2 * \ldots * m_r\text{，则同余方程}$$
$$\begin{cases} x \equiv b_1 \mod m_1 \\ x \equiv b_2 \mod m_2 \\ \ldots x \equiv b_r \mod m_r \end{cases}$$
$$\text{在模}N\text{同余的意义下有唯一解。}$$

由于m_i两两互质，令$x=(N/m_i)*y$，方程组等同于解同余方程$(N/m_i)y \equiv 1 \mod m_i$，得到特解$y_i$，则方程组的解为：$x_0 = b_1x_1 + b_2x_2 + \ldots + b_rx_r$

模板

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
//P1495 【模板】中国剩余定理(CRT)/曹冲养猪

long long n,mul=1,m[16],a[16],mi[16],x,ans=0;

void exgcd(ll a,ll b,ll &x,ll &y){
    if(!b){
        x=1;y=0;
        return;
    }
    exgcd(b,a%b,x,y);
    ll tmp=x;
    x=y;
    y=tmp-a/b*y;
}

int main(){
    scanf("%lld",&n); //同余方程个数
    for(int i=1;i<=n;i++){
        scanf("%lld%lld",&x,&a[i]); //余数和模数
        m[i]=x;
        mul*=x; //记录大模数N
    }
    for(int i=1;i<=n;i++){
        mi[i]=mul/m[i]; //
        ll x,y;
        exgcd(mi[i],m[i],x,y);
        ans+=a[i]*mi[i]*(x<0?x+m[i]:x);
    }
    printf("%lld",ans%mul);
    return (0^0);
}
```

总结步骤

求大模数M

对于每个小模数p=M/m,求模m意义下的逆元i，那么p*i*a就是满足方程的最小数

每个方程的最小数相加模M，就是方程组最小解。

其他

完全数（完美数）： 全部因数之和等于他本身

盈数： 全部因数之和大于他本身

亏数： 全部因数之和小于他本身

亲和数： 一个属的真因子的和等于另一个数

参考代码

luoguP3518 [POI2011]SEJ-Strongbox:https://www.luogu.com.cn/record/54555918

UVA374 Big Mod：https://www.luogu.com.cn/record/54557937

UVA11105 H-半素数 Semi-prime H-numbers：https://www.luogu.com.cn/record/54559621

UVA756 Biorhythms：https://blog.nowcoder.net/n/037e60b2ee124f6286e2b105b7e4c9bf

参考资料

信息学奥赛之数学一本通 （林厚从）

https://www.bilibili.com/video/BV1o5411T7Np?from=search&seid=3644460168410828006

数论入门——阮行止