> 1.运用CRT求解：
>
> $$x \equiv 8\ (mod\ 11)$$
>
> $$x \equiv\ 3(mod\ 19)$$

解：

该同余方程组存在唯一解
$x=8\times 19\times 19^{-1}+3\times 11\times 11^{-1}\pmod{11\times 19}$，其中$19^{-1}$为19在11下的乘法

逆元，11^{-1}为11

在19下的乘法逆元。

根据$egcd$：

$$\begin{pmatrix} 1 & 0 & 19 \\ 0 & 1 & 11 \\ 1 & -1 & 8 \\ 2 & -2 & 16 \\ 2 & -3 & 5 \\ -1 & 2 & 3 \\ -2 & 4 & 6 \\ -4 & 7 & 1 \end{pmatrix}$$

可得$19^{-1}=7,11^{-1}=7$，代入得：$x=41$

> 2．运用CRT求解：
>
> $$x \equiv 1\ (mod\ 5)$$
> $$x \equiv 2\ (mod\ 7)$$
> $$x \equiv 3\ (mod\ 9)$$
> $$x \equiv 4\ (mod\ 11)$$

解：

该同余方程组有唯一解$x=\sum_{i=0}^{3} a_i b_i b_i^{-1}\pmod{M}$，记
$M=5\times 7\times 9\times 11=3465$，$b_i=M/m_i$，其中

$m_0 = 5, m_1 = 7, m_2 = 9, m_3 = 11$，有：
$b_0 = 693, b_1 = 495, b_2 = 385, b_3 = 315$

记b_i^{-1}为b_i在m_i下的逆元。

根据$egcd$：

$$\begin{pmatrix} 1 & 0 & 693 \\ 0 & 1 & 5 \\ 0 & 138 & 690 \\ 1 & -138 & 3 \\ 2 & -276 & 6 \\ 2 & -277 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 495 \\ 0 & 1 & 7 \\ 0 & 70 & 490 \\ 1 & -70 & 5 \\ -1 & 71 & 2 \\ -2 & 142 & 4 \\ 3 & -212 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 385 \\ 0 & 1 & 9 \\ 0 & 42 & 378 \\ 1 & -42 & 7 \\ -1 & 43 & 2 \\ -3 & 129 & 6 \\ 4 & -171 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 315 \\ 0 & 1 & 11 \\ 0 & 28 & 308 \\ 1 & -28 & 7 \\ -1 & 29 & 4 \\ 2 & -57 & 3 \\ 8 & -228 & 12 \\ 8 & -229 & 1 \end{pmatrix}$$

故$b_0^{-1} = 2, b_1^{-1} = 3, b_2^{-1} = 4, b_3^{-1} = 8$

解得：$x = 1731$

> 3．手动计算$2000^{2019} \pmod{221}$。

解：

$221 = 11 \times 13$，$2000 \leftrightarrow (11, 11)$

即求$(11, 11)^{2019} = ([11^{2019}\ mod\ \ 17], [11^{2019}\ mod\ \ 13])$

由费马小定理得：

$$11^{2019} = 11^{126\times16+3} \equiv 1(mod\,17)$$
$$11^{2019} = 11^{168\times12+3} \equiv 1(mod\,13)$$

故有：
$([11^{2019}\ mod\ \ 17], [11^{2019}\ mod\ \ 13]) = ([11^3\ mod\ \ 17], [11^3\ mod\ \ 13]) = (5, 5)$
，又因Z_n与$Z_p \times Z_q$同

构，存在双射，故$(5, 5) \leftrightarrow 5$

故答案为5。

> 4.实现一个利用CRT求解同余方程的程序。

```
#define N 10
struct Bezout
{
    int r;
    int s; //r、s为Bezout系数
    int d; //d为gcd(a,b)
};
int crt(int a[N], int m[N], int d, int sum_m); //用CRT求解同余方程组函数
int Mul_Inverse(int a, int m); //求乘法逆元的函数
Bezout egcd(int a, int b);//求Bezout系数
int main()
{
    //d为这个方程组内的方程数
    int a[N], m[N], b[N], b_[N], d = 0, x = 0, sum_m = 1;
    for (int i = 0; i <= N; i++)
    {
        cout << "请输入方程组中每个方程的a与m，输入0停止。" << endl;
        cin >> a[i] >> m[i];
        if (a[i] != 0 && m[i] != 0)
        {
            sum_m *= m[i];
        }
        else
        {
            d = i;
            break;
        }
    }
    cout << "x=" << crt(a, m, d, sum_m) << endl;
    return 0;
}

int crt(int a[N], int m[N], int d, int sum_m)
{
    int b[N], b_[N], x = 0;
    for (int i = 0; i < d; i++)
    {
        b[i] = sum_m / m[i];
        b_[i] = Mul_Inverse(b[i], m[i]);
    }
```

```
    for (int i = 0; i < d; i++)
    {
        x += (a[i] * b[i] * b_[i]);
    }
    return (x % sum_m);
}
Bezout egcd(int a, int b) //求Bezout系数
{
    Bezout res;
    int r0 = 1, r1 = 0, s0 = 0, s1 = 01, q; //初始化
    while (b)
    {
        q = a / b;
        //以下计算gcd(a,b)
        int temp = a % b;
        a = b;
        b = temp;
        //以下计算Bezout系数
        int tr = r0;
        int ts = s0;
        r0 = r1;
        r1 = tr - q * r1;
        s0 = s1;
        s1 = ts - q * s1;
    }
    res.r = r0;
    res.s = s0;
    res.d = a;
    return res;
}
int Mul_Inverse(int a, int m)
{
    Bezout res;
    res = egcd(a, m);
    return res.r;
}
```