> 1. 群Z_{17}^*有多少个生成元？已知3是其中一个生成元，请问9和10是否生成元？

解：

根据**原根定理**：对每一个素数p都有模p的原根，且恰有$\phi(p-1)$个模p原根。

故Z_{17}^*有$\phi(17-1)=\phi(16)=16\times\dfrac{1}{2}=8$个原根。

使用群的语言，我们可知：群Z_{17}^*有8个生成元。

已知3是其中一个生成元：

$3^2=9\in Z_{17}^*$，$gcd(2,16)=2$，所以9的阶不是16，故9不是生成元。

$3^3=10\in Z_{17}^*$，$gcd(3,16)=1$，所以10的阶也为16，故10是生成元。

> 2. p和q是两个不同的素数，请问Z_{pq}都多少个生成元？r是任意正整数，请问Z_{p^r}都多少个生成元？

解：

设g为群Z_{pq}中的生成元，群Z_{pq}有pq个元素，都可表达为g^i，其中$i\in Z$。对任意元素$h=g^i\in Z_{pq}$，h的阶为pq/d，其中$d=gcd(i,pq)$。当$d=1$时，h的阶等于群的阶，即h为单位元。在群Z_{pq}中共有$\phi(pq)$个元素与pq互素。而$\phi(pq)=\phi(p)\phi(q)=(p-1)(q-1)$，故$Z_{pq}$中有$(p-1)(q-1)$个生成元。

同理：Z_{p^r}中共有$\phi(p^r)$个元素与p^r互素。$\phi(p^r)=p^r-p^{r-1}$

故Z_{p^r}中有p^r-p^{r-1}个生成元。

> 3. 证明：如果群G没有非平凡子群，则群G是循环群。

证明：

可证**逆否命题**：如果G不是循环群，则G必有非平凡子群。

任取非单位元元素$x\in G$，x经过群操作后必定会产生新的元素$y\in C$使得$C\subset G$，故C为G的非平凡子群。

证毕。

> 4．证明：设G为任意群，且$g \in G$。如果存在m，$n \in G$使得$g^m = 1$且$g^n = 1$，则$g^d = 1$，其中$d = gcd(m, n)$。

证明:

根据$Bezout$定理：$\exists r, s\ s.t.\ gcd(m, n) = mr + ns$，又根据题意有：

$$g^d = g^{gcd(m,n)} = g^{(mr+ns)} = (g^m)^r (g^n)^s = 1$$

证毕。

> 5．设G是群，H是G的子群。任取$g_1, g_2 \in G$，则$g_1 H = g_2 H$当且仅当$g_1^{-1} g_2 \in H$。

证明:

$\Rightarrow$:

由于$g_1 H = g_2 H$，故存在$h_1, h_2 \in H\ s.t.\ g_1 h_1 = g_2 h_2$，有：
$g_1 h_1 = g_1 g_1^{-1} g_2 h_2$

根据消去律与封闭性有：$g_1^{-1} g_2 = h_1 h_2^{-1} \in H$。

$\Leftarrow$:

任取$g_1 h \in g_1 H$，由于$g_1^{-1} g_2 \in H$，存在$h' \in H$使得$g_1^{-1} g_2 = h'$，故
$g_1 = g_2 (h')^{-1}$

$g_1 h = g_2 (h')^{-1} h \in g_2 H$，故$g_1 H \subseteq g_2 H$。

任取$g_2 h \in g_2 H$，由于$g_1^{-1} g_2 \in H$，存在$h' \in H$使得$g_1^{-1} g_2 = h'$，故
$g_2 = g_1 h'$

$g_2 h = g_1 h' h \in g_1 H$，故$g_2 H \subseteq g_1 H$

所以$g_2 H = g_1 H$，证毕。

> 6．如果G是群，H是群G的子群，且$[G : H] = 2$，请证明对任意的$g \in G$，$gH = Hg$。

证明:

由于H为G的子群且$[G : H] = 2$，则群G被划分成两个不同的左陪集。

$i.$ 任取$g \in H$，根据H的封闭性，有$gH = H = Hg$

$ii.$ 任取$g \notin H$，gH与Hg必会落在陪集$G - H$上，有$gH = G - H = Hg$

证毕。

> 7. 设G是阶为pq的群，其中p和q是素数。请证明G的任意非平凡子群是循环群。

证明：

根据拉格朗日定理，子群H的阶必然整除群G的阶。由于G的阶为pq，pq因子只有$1, p, q, pq$。又因为H是G的任意非平凡子群，其阶只能为p或q。素数阶的群是循环群，故G的任意非平凡子群是循环群。

下证素数阶的群都是循环群：

$\forall h \in H$，$ord(h) || H|$，又因为$|H|$位素数，故对于任意非单位元元素的阶只能为$|H|$，此时元素的阶等于群的阶，故对于素数阶群来说，任意非单位元元素都是它的生成元，故素数阶群为循环群。证毕。

> 8. 编程完成以下工作：对任意给定的一个素数p，求出Z_p^*的最小生成元。任取一个整数n，对大于1小于n的所有素数p，求Z_p^*的最小生成元，并求以上最小生成元集合中最大者所对应的素数p。

```
int gcd(int a, int b);
int power(int a, int b, int p); //模指数运算
int Zpmmd(int p); //求Z_p^*最小生成元
bool isPrime(int n); //判断素数
void Zn_pmmd(int n); //任取一个整数n，对大于1小于n的所有素数p，求Zp^*的最小生成元,并求以上最小生成元集合中最大者所对应的素数p
int maxarr(int arr[1000]); //求数组元素中的最大值

int main()
{
    int p, n;
    cout << "输入素数p: ";
    cin >> p;
    cout << "Z_p^* 的最小生成元为:" << Zpmmd(p) << endl;
    cout << "输入一个整数n:" << endl;
    cin >> n;
```

```
    Zn_pmmd(n);
    return 0;
}

int gcd(int a, int b)
{
    int r;
    r = a % b;
    if (r == 0) return b;
    else return gcd(b, r);
}
int power(int a, int b, int p)
{
    int res = 1;
    while (b > 0)
    {
        if ((b & 1) == 1)
        {
            res = (res * a) % p;
        }
        b /= 2;
        a = (a * a) % p;
    }
    return res;
}

int Zpmmd(int p)
{
    for (int i = 1; i < p; i++)//在Z_p中遍历寻找最小生成元
    {
        int count = 0;
        int d = 0; //因子个数
        for (int j = 1; j < p; j++) //寻找p-1的因子f
        {

            if (gcd(j, p - 1) != 1) continue;
            else
            {
                d += 1;
                //j为因子时
                if ((power(i, j, p)) == 1) continue;
                else
```

```
                {
                    count++;
                }
            }

        }
        if (count == d) return i;
    }
}

bool isPrime(int n)
{
    if (n <= 1) return false;
    if (n == 2 || n == 3) return true;
    for (int i = 2; i <= int(sqrt(n)); i++)
    {
        if (n % i == 0)
        {
            return false;
        }
    }
    return true;
}

int maxarr(int arr[1000])
{
    int max = 0;
    for (int i = 0; i < 1000; i++)
    {
        if (max < arr[i]) max = arr[i];
    }
    return max;
}

void Zn_pmmd(int n)
{
    int p[1000], Z_p[1000], c = 0, d = 0, P[1000]; //大于1小于n的所有素数组成的集合
    for (int i = 2; i < n; i++)
    {
        if (isPrime(i))
        {
```

```
                d += 1; //用于记录素数个数
                p[i - 2] = i;
            }
        }
        //将上面所有素数所对应的最小生成元放入Z_p[]中
        for (int i = 0; i < d; i++)
        {
            Z_p[i] = Zpmmd(p[i]);
        }
        //此时最大的最小生成元为maxarr(Z_p)
        for (int i = 0; i < d; i++)
        {
            if ((Zpmmd(p[i])) == maxarr(Z_p))
            {
                P[c] = p[i];
                c += 1;//用于记录最大最小生成元对应的素数p的个数
            }
        }
        cout << "最大的最小生成元对应的素数为: ";
        for (int i = 0; i < c; i++)
        {
            cout << P[i] << " ";
        }
}
```