原根

模m有原根的充要条件：

$$m=2,4,p^a,2p^a(\text{其中}p\text{是奇素数})$$

一个数m如果有原根，则其原根个数为phi(phi(m))；特别地，对素数有phi(p)=p-1。

假设g是奇素数p的一个原根，则

$$g^1,g^2,\ldots,g^{p-1}$$

在模p意义下两两不同，且结果恰好为1~p-1，由此可以定义"离散对数"，与连续数学中的对数有异曲同工之妙。

离散对数又叫做"指标"，有指标法则：

$$I(ab)\equiv I(a)+I(b)(\%(p-1))$$

$$I(a^k)\equiv k*I(a)(\%(p-1))$$

由此可以把乘法转化为加法。

2017 revenge

题意：给你 n（2e6）个数，和一个数r，问你有多少种方案，使得你取出某个子集，能够让它们的乘积 mod 2017等于r。

题解：2017有5这个原根，可以使用离散对数（指标）的思想把乘法转化成加法。

$$\text{考虑}dp[i][j]\text{表示到第}i\text{个点后取模结果为}j\text{的数的个数}$$

$$f(i,j)=f(i-1,j)+f(i-1,(j-I(a(i))\ );I(i)\text{规定为}i\text{的指标}$$

```
#include <bits/stdc++.h>
using namespace std;
int I[2020];
int main(){
    bitset<2016>f;
    int xx=1;
    for (int i=0;i<2016;i++){
        I[xx]=i;
        xx=(xx*5)%2017;
    }
    int n,r;
    while(~scanf("%d%d",&n,&r)){
        f.reset();
        f.set(I[1]);
        for (int i=1;i<=n;i++){
            int x;
            cin>>x;
            f^=((f<<I[x]) ^ (f>>(2016-I[x])));
        }
        cout<<f[I[r]]<<endl;
    }
    return 0;
```

```
}
```