1117 - Helping Cicada

Given a number N and M integers. You have to calculate the number of integers in the range from 1 to N which are not divisible by any of those M integers.

Each case contains two integers N($1 \le N < 2^{31}$) and M($1 \le M \le 15$). The next line contains M positive integers(32 bit signed integer).

Sample Code:

```
#define ll long long
ll n,k,a[20];
ll LCM(ll mask){
    ll lcm=1;  int i=0;
    while(mask!=0){
        int bit = mask%2;
        if(bit==1){
            ll gcd = __gcd(lcm,a[i]);
            lcm = (lcm*a[i])/gcd;
            if(lcm>n)return n+1;
        }
        mask /= 2; i++;
    }
    return lcm;
}
ll solve(){
    ll cnt=0;
    for(ll mask=1; mask<(1<<k); mask++){
        ll bit = __builtin_popcountll(mask);
        ll lcm = LCM(mask);
        if(bit%2==1)cnt+=(n/lcm);
        else cnt-=(n/lcm);
    }
    return n-cnt;
}
int main()
{
    int t; scanf("%d",&t);
    for(int ks=1; ks<=t; ks++){
        scanf("%d%d",&n,&k);
        for(int i=0; i<k; i++)scanf("%d",&a[i]);
        ll ans = solve();
        printf("Case %d: %lld\n",ks,ans);
    }
    return 0;
}
```