A - 筛1

```
#include<iostream>
#include<cstring>
#include<algorithm>
using namespace std;
int a[1000005];
int main(){
    int t,j;
    cin>>t;
    for(int i=1;i<=t;i++){
        int sum,cnt;
        cin>>sum>>cnt;
        for(j=1;j<=cnt;j++){
            cin>>a[j];
        }
        sort(a+1,a+1+cnt);
        int ans=0;
        bool flag=0;
        for(j=cnt;j>=1;j--){
            ans+=a[j];
            if(ans>=sum){
                flag=1;
                break;
            }
        }
        cout<<"Scenario #"<<i<<":"<<endl;
        if(flag) cout<<cnt-j+1<<endl<<endl;
        else cout<<"impossible"<<endl<<endl;
    }
    return 0;
}
```

B - 筛2

```
#include <iostream>
#include <cstdio>
#include <cstring>
#include <cmath>
#include <algorithm>
#include <vector>
#include <map>
using namespace std;
const int N = 1040400;
int vis[N], prime[N], num[N][3];
char str[N];
int main() {
    memset(vis,0,sizeof(vis));
    int k=0;
    int t=(int)sqrt(N*1.0);
    for(int i=2;i<=t;i++) {
        if(vis[i]) continue;
        if(i%2==0) continue;
        for(int j=2*i;j<N;j+=i) vis[j]=1;
    }
    for(int i=2;i<N;i++)
        if(!vis[i]) prime[k++]=i;
    int n;
    while(scanf("%s %d",str,&n)!=EOF&&n) {
        if(str[0]=='0'&&n==0) break;
        int len=strlen(str), cnt=0;
        for(int i=0;str[i]!='\0';i++) {
            if(i+2<len) {
                num[cnt][1]=1000;
                num[cnt++][0]=(str[i]-'0')*100+(str[i+1]-'0')*10+(str[i+2]-'0');
                i+=2;
            }
            else if(i+1<len) {
                num[cnt][1]=100;
                num[cnt++][0]=(str[i]-'0')*10+(str[i+1]-'0');
                i+=1;
            }
            else if(i<len) {
                num[cnt][1]=10;
                num[cnt++][0]=(str[i]-'0');
            }
        }
        int flag=0;
        for(int i=0;i<k;i++) {
            if(prime[i]>=n) break;
            int sum=0;
            for(int j=0;j<cnt;j++) {
                sum=sum*num[j][1]+num[j][0];
                sum%=prime[i];
            }
            if(sum==0) {
                printf("BAD %d\n",prime[i]);
                flag=1;
                break;
            }
        }
        if(!flag)puts("GOOD");
    }
    return 0;
}
```

C - Extended Twin Composite Number

```
#include<iostream>
#include<cstdio>
#include<cstring>
#include<algorithm>
using namespace std;
const int Mod = 1000000007;
int main(){
    int line;
    cin >> line;
    while ( line--) {
        long long num;
        scanf("%lld",&num);
        long long temp1 = 8;
        long long temp2 = 9;
        printf("%lld %lld\n",num*temp1, num*temp2);
    }
    return 0;
}
```

D - 快速幂

```
#include<iostream>
#include<cstdio>
using namespace std;
typedef long long ll;
const int N=1000000007;
ll pow(ll a,ll n,ll p) {
    ll ans = 1;
    while(n) {
        if(n & 1 ) ans = (ans * a) % p;
        a = a * a % p;
        n >>= 1;
    }
    return ans;
}
int main() {
    int n;
    while(~scanf("%lld",&n)) {
        cout<<pow(2,n*n-1,N)<<endl;
    }
    return 0;
}
```

E - 矩阵快速幂

```
#include<iostream>
#include<stdio.h>
#include<string.h>
using namespace std;
struct juzhen{
    int v[2][2];
};
juzhen t={1,1,1,0};
int tmp[2][2];
juzhen multi(juzhen a,juzhen b,int n) {
    memset(tmp,0,sizeof tmp);
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            for(int k=0;k<n;k++)
                tmp[i][j]=(tmp[i][j]+a.v[i][k]*b.v[k][j])%10000;
    for(int i=0;i<n;i++)
        for(int j=0;j<n;j++)
            a.v[i][j]=tmp[i][j];
    return a;
}
juzhen res;
int Pow(juzhen a,long long n) {
    res.v[0][0]=res.v[1][1]=1;
    res.v[0][1]=res.v[1][0]=0;
    while(n){
        if(n&1)
            res=multi(res,a,2);
        a=multi(a,a,2);
        n>>=1;
    }
    return res.v[0][1];
}

int main(){
    long long n;
    while(scanf("%lld",&n)!=EOF&&n!=-1){
        if(n==0) cout<<0<<endl;
        else
            printf("%d\n",Pow(t,n)%10000);
    }
    return 0;
}
```

F - 欧拉函数

```
#include<iostream>
#include<cmath>
using namespace std;
int euler_phi(int n) {
    int m = int(sqrt(n + 0.5));
    int ans = n;
    for (int i = 2; i <= m; i++)
        if (n % i == 0) {
            ans = (ans / i) * (i - 1);
            while (n % i == 0) n /= i;
        }
    if (n > 1) ans = (ans / n) * (n - 1);
    return ans;
}
int main(){
    int t;
    cin>>t;
    for(int i=1;i<=t;i++){
        int n;
        cin>>n;
        long long sum=0;
        for(int j=2;j<=n;j++)
            sum+=euler_phi(j);
        sum=sum*2+3;

        cout<<i<<' '<<n<<' '<<sum<<endl;
    }
    return 0;
}
```

G - 欧拉降幂

```
#include<iostream>
#include<cstring>
```

```
#include<cstdio>
using namespace std;
typedef long long ll;
char b[1000006];
ll a,c;
ll qmod(ll a,ll n,ll mod){
    ll ans=1;
    while(n){
        if(n&1){
            ans=ans*a%mod;
        }
        a=a*a%mod;
        n>>=1;
    }
    return ans%mod;
}
ll euler(ll n){
    ll ans=n;
    for(ll i=2;i*i<=n;i++){
        if(!(n%i)){
            ans=ans/i*(i-1);
            while(n%i==0){
                n/=i;
            }
        }
    }
    if(n>1) ans=ans/n*(n-1);
    return ans;
}
int main(){
    while(~scanf("%lld %s %lld",&a,b,&c)){
        ll PHI=euler(c);
        ll len=strlen(b);
        ll res=0;
        for(ll i=0;i<len;i++){
            res=(res*10+b[i]-'0')%PHI;
        }
        printf("%lld\n",qmod(a,res,c));
    }
    return 0;
}
```

H - 组合数

```
#include<iostream>
#include<string.h>
#include<cstdio>
using namespace std;
#define ll long long
ll p;
inline ll qpow(ll a,ll b) {
    if(b==1) return a;
    ll res=1;
    while(b)
    {
        if(b&1)
            res=(res*a)%p;
        a=(a*a)%p;
        b/=2;
    }
    return res;
}
inline ll C(ll n,ll m) {
    if(n<m) return 0;
    if(m>n-m) m=n-m;
    ll a=1,b=1;
    for(int i=0; i<m; i++) {
        a=(a*(n-i))%p;
        b=(b*(i+1))%p;
    }
    return a*qpow(b,p-2)%p;
}
inline ll Lucas(ll n,ll m) {
    if(m==0) return 1;
    return Lucas(n/p,m/p)*C(n%p,m%p)%p;
}
int main() {
    int T;
    scanf("%d",&T);
    ll n,m;
    while(T--)
    {
        scanf("%I64d%I64d%I64d",&n,&m,&p);
        ll ans=Lucas(n+m,n);
        printf("%I64d\n",ans);
    }
    return 0;
}
```

I - OEIS.org

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const ll Mod=998244353;
ll a[100005];
void init(){
    a[0]=a[4]=a[5]=1;
    a[2]=a[1]=0;
    a[5]=8;
    a[6]=36;
    for(int i=7;i<=100003;i++){
        a[i]=((i-2)*a[i-1]%Mod+(i-1)*a[i-2]%Mod+(i&1?1:-1))%Mod;
    }
    return ;
}
int main(){
    int t;
    cin>>t;
    init();
    while(t--){
        int n;
        cin>>n;
        cout<<a[n]<<endl;
    }
    return 0;
}
```

J - 扩展欧几里得

```
#include<iostream>
#include<cmath>
#define ll long long
using namespace std;
ll  a,b,m,n,l,x,y;
ll exgcd(ll a,ll b,ll &x,ll &y ) {
    if(b==0) {
        x=1;
        y=0;
        return a;
    }
    ll d=exgcd(b,a%b,x,y);
    ll z=x;
    x=y;
    y=z-a/b*y;
    return d;
}
int main() {
    cin>>a>>b>>m>>n>>l;
    ll d=exgcd(m-n,l,x,y);
    if((b-a)%d) {
        cout<<"Impossible";
    } else {
        x*=(b-a)/d;
        ll mod;
        if(l/d<0) mod=-(l/d);
        else mod=l/d;
        cout<<(x%mod+mod)%mod;
    }
    return 0;
}
```

K - CRT(中国剩余定理)

```
#include<iostream>
#include<stdio.h>
using namespace std;
int exgcd(int a,int b,int &x,int &y) {
    if(b == 0 ) {
    x = 1;
    y = 0;
    return a;
    }
    int d = exgcd(b, a%b , x, y);
    int z = x ;
    x = y;
    y = z - y*(a/b);
    return d;
}
int CRT(int A[],int B[],int n){
    int m = 1,ans = 0;
    for(int i = 0;i < n;++i)
        m = m * B[i];
    for(int i = 0;i < n;++i){
        int x,y;
        int Mi = m / B[i];
        exgcd(Mi,B[i],x,y);
        ans = (ans + Mi * x * A[i]) % m;
    }
    if(ans <= 0) ans += m;
    return ans;
}
int main(){
    int d;
    int b[3]={23,28,33};
    for(int i=1;i;i++){
        int a[3];
        cin>>a[0]>>a[1]>>a[2]>>d;
        if(a[0]+a[1]+a[2]+d==-4) break;
        int sum=CRT(a,b,3);
        if(sum<d) sum+=21252-d;
        else sum-=d;
        printf("Case %d: the next triple peak occurs in %d days.\n",i,sum);
    }
    return 0;
}
```

L - EXCRT1（扩展中国剩余定理）

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#define MOD 1e9+7
#define ll long long
#define ULL unsigned long long     //1844674407370955161
#define INT_INF 0x7f7f7f7f      //2139062143
#define LL_INF 0x7f7f7f7f7f7f7f7f //9187201950435737471
using namespace std;
ll exgcd(ll a,ll b,ll &x,ll &y) {
    if(b == 0 ) {
        x = 1;
        y = 0;
        return a;
    }
    ll d = exgcd(b, a%b , x, y);
    ll z = x ;
    x = y;
    y = z - y*(a/b);
    return d;
}
//逐一合并大法
ll CRT(int w[],int b[],int k)//w为除数，b为余数，k为有多少式子
{
    ll Wi=w[0],ret=b[0];
    for(int i=1;i<k;++i) {
        ll wi=w[i];ll bi=b[i];ll x,y;
        ll gcd=exgcd(Wi,wi,x,y);ll c=bi-ret;
        if(c%gcd)//表示没有结果
            return -1;
        ll W=wi/gcd;
        ret+=wi*(((c/gcd*x)%W+W)%W);
        wi*=W;
    }
    if(!ret)//表示余数全部为零
    {
        ret=1;
        for(int i=0;i<k;++i)
            ret=ret*w[i]/__gcd(ret,(ll)w[i]);//使用库函数求最小公倍数
    }
    return ret;
}
int main()
{
    int w[10000],b[10000];
    int n;
    while(scanf("%d",&n)!=EOF) {
        for(int i=0;i<n;++i) {
            scanf("%d%d",&w[i],&b[i]);//除数
        }
        ll ans=CRT(w,b,n);
        printf("%lld\n",ans);
    }
    return 0;
}
```

M - EXCRT2

```
#include<iostream>
#include<cstdio>
#include<algorithm>
#define MOD 1e9+7
#define ll long long
#define ULL unsigned long long     //1844674407370955161
#define INT_INF 0x7f7f7f7f      //2139062143
#define LL_INF 0x7f7f7f7f7f7f7f7f //9187201950435737471
using namespace std;
ll exgcd(ll a,ll b,ll &x,ll &y) {
    if(b == 0 ) {
        x = 1;
        y = 0;
        return a;
    }
    ll d = exgcd(b, a%b , x, y);
    ll z = x ;
    x = y;
    y = z - y*(a/b);
    return d;
}
//逐一合并大法
ll lcm = 0;
ll CRT(ll w[],ll b[],ll k)//w为除数，b为余数，k为有多少式子
{
    ll wi=w[0],ret=b[0];
    for(int i=1;i<k;++i) {
        ll wi=w[i];ll bi=b[i];ll x,y;
        ll gcd=exgcd(Wi,wi,x,y);ll c=bi-ret;
        if(c%gcd)//表示没有结果
            return -1;
        ll W=wi/gcd;
        ret+=Wi*(((c/gcd*x)%W+W)%W);
        Wi*=W;
    }
    lcm = Wi;
    if(!ret)//表示余数全部为零
    {
        ret=1;
        for(int i=0;i<k;++i)
            ret=ret*w[i]/__gcd(ret,(ll)w[i]);//使用库函数求最小公倍数
        lcm = ret;
    }
    return ret;
}
int main() {
    int t;
    cin >> t;
    while(t--) {
        int n,m;
        cin >> n >> m;
        ll a[15],b[15];
        for(int i = 0 ; i < m ;i++) {
            cin >> a[i];
        }
        for(int i = 0 ; i < m ;i++) {
            cin >> b[i];
        }
        ll ans=CRT(a,b,m);
        if (ans==-1||ans>n)
            cout<<"0"<<endl;
        else
            cout<<(n-ans)/lcm+1<<endl;

    }
    return 0;
}
```