A.

题意:

求$\sum_{i=k}^{n} C_n^i$

题解:

由于n很大，不能直接求。由公式$\sum_{i=0}^{n} C_n^i = 2^n$,所以$\sum_{i=k}^{n} = 2^n - \sum_{i=0}^{k-1} C_n^i$

但是还有个问题，就是n太大不能预处理阶乘，不能快速求组合数。由公式$C_n^k = \frac{n-k+1}{k} C_n^{k-1}$ 一个一个递推即可。

B.

题意:

给了一个数组a ,有m次操作，每次操作给一个k, 然后令$b_i = \sum_{j=i-k\cdot x} a_j (0 \le x, 1 \le j \le i)$，再把$b$数组复制回$a$ 数组，问经过k次操作后a的数组值

题解:

为了方便起见，让下标从0开始。考虑生成函数$A(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$

$$\begin{aligned} B(x) &= \sum_{i \ge 0} b_i x^i = \sum_{i \ge 0} \sum_{\substack{j=i-kz \\ j \ge 0, z \ge 0}} a_j x^j \cdot 1 \cdot x^{i-j} \\ &= \sum_{i \ge 0} \sum_{0 \le z \le \lfloor \frac{i}{k} \rfloor} a_{i-kz} x^{i-kz} \cdot 1 \cdot x^{kz} \\ &= \sum_{i \ge 0} \sum_{j=0} a_{i-j} x^{i-j} \cdot [k|j] \cdot x^j \quad \text{这里}[\text{k}|\text{j}]\text{表示当k能整除j时值才为}1\text{，否则是}0 \\ &= A(x) * (1 + x^k + x^{2k} + \cdots) \end{aligned}$$

也就是说一次操作相当于进行一次多项式乘法，而众所周知乘法是由交换律的，所以操作的顺序并没有影响

用$cnt[1], cnt[2], cnt[3]$分别表示各个操作进行了几次，所以答案就是

$$A(x) * (1 + x + x^2 + x^3 + \cdots)^{cnt[1]} * (1 + x^2 + x^4 + \cdots)^{cnt[2]} * (1 + x^3 + x^6 + \cdots)^{cnt[3]}$$

实际上我们并不需要超过n次项，现在只要分别求出各个多项式的cnt 次方，再最后套一个NTT板子就完事了。

然而显然不能直接把多项式乘cnt 次，就算用FFT优化也是$O(mn\log n)$（如果你FFT以后，对于每一项单独普通快速幂是不对的，因为FFT乘法需要预处理到结果的最高此项，也就是nm次）

于是就有多项式快速幂，类似普通快速幂的做，只不过是用多项式乘法而已，注意要清0大于n次方项的系数（并没有真正用多项式快速幂做过，口胡的一个做法，可能被卡掉）这样复杂度是$O(n\log n\log m)$

注意到$1 + x^k + x^{2k} + \cdots = \frac{1}{1-x^k}$，于是:

$$
\begin{aligned}
(1+x^k+x^{2k}+\cdots)^m &= \frac{1}{(1-x^k)^m}\\
&=(1-x^k)^{-m}\\
&=\sum_{i=0}^{\infty}(-1)^i\binom{m+i-1}{i}(-x^k)^i\\
&=\sum_{i=0}^{\infty}\binom{m+i-1}{i}x^{ki}
\end{aligned}
$$

然后用这个式子做3次多项式乘法就行了

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;typedef double db;
typedef pair<int, int> pii;typedef pair<ll, ll> pll;
typedef pair<int,ll> pil;typedef pair<ll,int> pli;
#define Fi first
#define Se second
#define _Out(a) cerr<<#a<<" = "<<(a)<<endl
const int INF = 0x3f3f3f3f, MAXN = 3e6 + 50;
const ll LINF = 0x3f3f3f3f3f3f3f3f, MOD = 998244353;
const db Pi = acos(-1), EPS = 1e-6;
void test(){cerr << "\n";}template<typename T,typename...Args>void test(T
x,Args...args){cerr<<x<<" ";test(args...);}
inline ll qpow(ll a, ll b){return b?((b&1)?
a*qpow(a*a%MOD,b>>1)%MOD:qpow(a*a%MOD,b>>1))%MOD:1;}
inline ll qpow(ll a, ll b,ll c){return b?((b&1)?
a*qpow(a*a%c,b>>1,c)%c:qpow(a*a%c,b>>1,c)) %c:1;}
inline ll gcd(ll a,ll b){return b?gcd(b,a%b):a;}
inline ll cede(ll a,ll b){if(b<0)return cede(-a,-b);if(a<0)return a/b;return
(a+b-1)/b;}
inline ll flde(ll a,ll b){if(b<0)return flde(-a,-b);if(a<0)return (a-
b+1)/b;return a/b;}
inline int sign(db x){return x<-EPS ? -1:x>EPS;}
inline int dbcmp(db l,db r){return sign(l - r);}
namespace Fast_IO{ //orz laofu
    const int MAXL((1 << 18) + 1);int iof, iotp;
    char ioif[MAXL], *ioiS, *ioiT, ioof[MAXL],*iooS=ioof,*iooT=ioof+MAXL-
1,ioc,iost[55];
    char Getchar(){
        if (ioiS == ioiT){
            ioiS=ioif;ioiT=ioiS+fread(ioif,1,MAXL,stdin);return (ioiS == ioiT ?
EOF : *ioiS++);
        }else return (*ioiS++);
    }
    void Write(){fwrite(ioof,1,iooS-ioof,stdout);iooS=ioof;}
    void Putchar(char x){*iooS++ = x;if (iooS == iooT)Write();}
    inline int read(){
        int x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return
x*iof;
    }
    inline long long read_ll(){
        long long x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
```

```cpp
        if(ioc==EOF)Write(),exit(0);
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return x*iof;
    }
    void Getstr(char *s, int &l){
        for(ioc=Getchar();ioc==' '||ioc=='\n'||ioc=='\t';)ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(l=0;!(ioc==' '||ioc=='\n'||ioc=='\t'||ioc==EOF);ioc=Getchar())s[l++]=ioc;s[l] = 0;
    }
    template <class Int>void Print(Int x, char ch = '\0'){
        if(!x)Putchar('0');if(x<0)Putchar('-'),x=-x;while(x)iost[++iotp]=x%10+'0',x/=10;
        while(iotp)Putchar(iost[iotp--]);if (ch)Putchar(ch);
    }
    void Putstr(const char *s){for(int i=0,n=strlen(s);i<n;++i)Putchar(s[i]);}
} // namespace Fast_IO
using namespace Fast_IO;
namespace DXS
{
    const int G = 3, Gi = 332748118;
    int bit=0,lim=1;
    int rev[MAXN];
    int init(int N){//初始长度
        lim=1,bit=0;while(lim<N)lim<<=1,bit++;
        for(int i=0;i<lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));
        return lim;
    }
    void NTT(int *A,int lim,int inv){
        for(int i=0;i<lim;i++)if(i<rev[i])swap(A[i],A[rev[i]]);
        for(int mid=1;mid<lim;mid<<=1){
            int W=qpow(inv==1?G:Gi,(MOD-1)/(mid<<1));
            for(int j=0;j<lim;j+=(mid<<1)){
                int w=1;for(int k=0;k<mid;k++,w=1ll*w*W%MOD){
                    int x=A[j+k],y=1ll*w*A[j+k+mid]%MOD;
                    A[j+k]=(x+y)%MOD;A[j+k+mid]=(x-y+MOD)%MOD;
                }
            }
        }
        if(inv==-1)for(int i=0,iv=qpow(lim,MOD-2);i<lim;i++)A[i]=1ll*iv*A[i]%MOD;
    }
    void Getdev(int *A,int *B,int n){for(int i=0;i<n-1;++i)B[i]=1ll*A[i+1]*(i+1)%MOD;}
    void Getint(int *A,int *B,int n){for(int i=n;i>=1;--i)B[i]=1ll*A[i-1]*qpow(i,MOD-2)%MOD;B[0]=0;}
    void Rightshift(int *A,int *B,int n,int k){for(int i=n-1;i>=k;--i)B[i]=A[i-k];for(int i=0;i<min(k,n);i++)B[i]=0;}
    void Leftshift(int *A,int *B,int n,int k){for(int i=n-k-1;i>=0;--i)B[i]=A[i+k];for(int i=n-1;i>=max(0,n-k);i--)B[i]=0;}
    void mulC(int *A,int n,int c){for(int i=0;i<n;i++)A[i]=1ll*A[i]*c%MOD;}
    void mul(int *A,int *B,int *C,int n,int m){//NOTE: A[i] and B[i] must be 0 when i>n, i>m (until 2n,2m)
        int len=init(n+m);NTT(A,len,1);NTT(B,len,1);
        for(int i=0;i<len;i++)C[i]=1ll*A[i]*B[i]%MOD;NTT(C,len,-1);
    }
    //int tmpA[MAXN];
    //void Getinv(int *A,int *B,int n)// return B;
```

```
    //{
    //  int now=1,len=2;
    //  for(int i=0;i<=n+n;i++)B[i]=0;
    //  B[0]=qpow(A[0],MOD-2);
    //  while(now<n+n){
    //      for(int i=0;i<now;i++)tmpA[i]=A[i];
    //      for(int i=now;i<len;i++)tmpA[i]=0;
    //      init(len);NTT(tmpA,len,1);NTT(B,len,1);
    //      for(int i=0;i<len;i++)B[i]=1ll*B[i]*(2-
1ll*tmpA[i]*B[i]%MOD+MOD)%MOD;
    //      NTT(B,len,-1);for(int i=now;i<len;i++)B[i]=0;
    //      now<<=1;len<<=1;
    //  }
    //}
    //int invA[MAXN],devA[MAXN];
    //void Getln(int *A,int *B,int n){
    //  for(int i=0;i<n+n;i++)invA[i]=devA[i]=0;
    //  Getinv(A,invA,n);Getdev(A,devA,n);
    //  mul(invA,devA,invA,n,n);Getint(invA,B,n);
    //  B[n]=0;
    //}
    //int lnB[MAXN];
    //void Getexp(int *A,int *B,int n)
    //{
    //  int now=1,len=2;
    //  for(int i=0;i<=n+n;i++)B[i]=0;
    //  B[0]=1;
    //  while(now<n+n)
    //  {
    //      for(int i=0;i<len;i++)lnB[i]=0;
    //      Getln(B,lnB,now);
    //      for(int i=0;i<now;i++)lnB[i]=(MOD-lnB[i]+A[i]+(i==0))%MOD;
    //      init(len);NTT(B,len,1);NTT(lnB,len,1);
    //      for(int i=0;i<len;i++)B[i]=1ll*lnB[i]*B[i]%MOD;
    //      NTT(B,len,-1);for(int i=now;i<len;i++)B[i]=0;
    //      now<<=1;len<<=1;
    //  }
    //}
    //int tmpA2[MAXN],lnA[MAXN];


    /*void GetSqrt(int *A,int *B,int n)
    {
        int k=qpow(2,MOD-2);
        for(int i=0;i<n+n;i++)tmpA[i]=0;
        int cnt0=0;while(A[cnt0]==0&&cnt0<n)cnt0++;
        int xs=A[cnt0],invxs=qpow(xs,MOD-2);
        xs=ECSY::solve(xs,MOD);
        xs=min(xs,(int)MOD-xs);
        Leftshift(A,tmpA2,n,cnt0);
        mulC(tmpA2,n,invxs);
        Getln(tmpA2,lnA,n);
        mulC(lnA,n,k);
        Getexp(lnA,B,n);
        Rightshift(B,B,n,min((ll)INF,1ll*cnt0/2));
        mulC(B,n,xs);
    }*/
}
```

```
int fac[MAXN],invfac[MAXN];
void init()
{
    fac[0]=1;for(int i=1;i<MAXN;i++)fac[i]=1ll*fac[i-1]*i%MOD;
    invfac[MAXN-1]=qpow(fac[MAXN-1],MOD-2);
    for(int i=MAXN-2;i>=0;i--)invfac[i]=1ll*invfac[i+1]*(i+1)%MOD;
}
int C(int n,int m)
{
    if(m<0||m>n)return 0;
    return 1ll*fac[n]*invfac[m]%MOD*invfac[n-m]%MOD;
}
int a[MAXN],poly[3][MAXN];
int cnt[4];
void work()
{
    int n=read(),m=read();memset(cnt,0,sizeof cnt);
    memset(a,0,sizeof a);memset(poly,0,sizeof poly);
    for(int i=0;i<n;i++)a[i]=read();
    for(int i=1;i<=m;i++){
        int x=read();x--;cnt[x]++;
    }
    for(int k=1;k<4;k++)for(int i=0;i<n;i++)poly[k-1][i]=(i%k?0:C(cnt[k-1]+i/k-
1,i/k));
    poly[0][0]=poly[1][0]=poly[2][0]=1;
    int len = DXS::init(n+n+n+1);DXS::NTT(a,len,1);
    for(int i=0;i<3;i++)DXS::NTT(poly[i],len,1);
    for(int i=0;i<len;i++)a[i]=1ll*a[i]*poly[0][i]%MOD*poly[1][i]%MOD*poly[2]
[i]%MOD;
    DXS::NTT(a,len,-1);
    ll ans=0;
    for(int i=0;i<n;i++)
        ans^=(1ll*a[i]*(i+1));
    printf("%lld\n",ans);
}
int main(){
    init();int T=read();for(int cas=1;cas<=T;cas++)work();
}
```

C.

题意:

有一个正m边形房子，每一面墙有n^2个位置可以涂颜色，总共有c种颜色，问有多少种本质不同的涂色方案，两种方案是等价的当可以通过旋转就能相等

题解:

稍微需要动一下脑的polya计数问题。因为经过旋转后，一面墙之间的元素是不会发生变化的，所以可以当作一面墙只能涂一种颜色，而颜色的种类数是c^{n^2}个，这样就变成最基本的polya计数问题了。这里直接给出答案，基本的polya求解过程在D题中

$$ans = \sum_{i=0}^{m-1} c^{n*n*gcd(m,i)}$$

```cpp
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;typedef double db;
typedef pair<int, int> pii;typedef pair<ll, ll> pll;
typedef pair<int,ll> pil;typedef pair<ll,int> pli;
#define Fi first
#define Se second
#define _Out(a) cerr<<#a<<" = "<<(a)<<endl
const int INF = 0x3f3f3f3f, MAXN = 1e6 + 50;
const ll LINF = 0x3f3f3f3f3f3f3f3f, MOD = 1e9+7;
const db Pi = acos(-1), EPS = 1e-1;
void test(){cerr << "\n";}template<typename T,typename...Args>void test(T
x,Args...args){cerr<<x<<" ";test(args...);}
inline ll qpow(ll a, ll b){return b?((b&1)?
a*qpow(a*a%MOD,b>>1)%MOD:qpow(a*a%MOD,b>>1))%MOD:1;}
inline ll qpow(ll a, ll b,ll c){return b?((b&1)?
a*qpow(a*a%c,b>>1,c)%c:qpow(a*a%c,b>>1,c)) %c:1;}
inline ll gcd(ll a,ll b){return b?gcd(b,a%b):a;}
void work()
{
    int n,m,c;
    scanf("%d%d%d",&n,&m,&c);
    ll ans=0;
    for(int i=0;i<m;i++)ans=(ans+qpow(c,1ll*n*n*gcd(m,i)))%MOD;
    printf("%lld\n",ans*qpow(m,MOD-2)%MOD);
}
int main()
{
    work();
}
```

D.

题意:

一个大小为n，可以翻转的正多边形，给顶点染色，问有多少种本质不同的染色方案，当可以通过任意次旋转/对称（翻转）相等的两种方案等价

题解:

polya入门题，首先要确定置换群。

旋转有n个：顺势针旋转$\frac{360i}{n}$度，$0 \le i < n$,这样第i个置换就有$\gcd(n,i)$个环

对称：

当n是奇数时，有n个对称轴——过第i个点和它对边的中点，这样每个置换都有$\lceil \frac{n}{2} \rceil$个环

当n时偶数时，也有n个对称轴——过第i和$i+\frac{n}{2}$个点（有$\frac{n}{2}+1$个环）或者过两条对边的中点（有$\frac{n}{2}$）

然后就奇偶讨论一下就是答案了

$$ans = (\sum_{i=0}^{n-1} 3^{\gcd(n,i)} + \begin{cases} n \cdot 3^{\frac{n}{2}+1}, n\%2 = 1 \\ \frac{n \cdot 3^{\frac{n}{2}} + n \cdot 3^{\frac{n}{2}+1}}{2}, n\%2 = 0 \end{cases}) \cdot \frac{1}{n}$$

```
#include<stdio.h>
typedef long long ll;
const ll LINF=1e18;
inline ll qpow(ll a, ll b){return b?((b&1)?a*qpow(a*a,b>>1):qpow(a*a,b>>1)) :1;}
inline ll gcd(ll a,ll b){return b?gcd(b,a%b):a;}
void work()
{
    int n;
    while(1)
    {
        scanf("%d",&n);
        if(n==-1)return;
        if(n==0)
        {
            printf("0\n");continue;
        }
        ll ans=0;
        for(int i=0;i<n;i++)
        {
            ans=(ans+qpow(3,gcd(n,i)));
        }
        if(n%2)
        {
            ans+=n*qpow(3,n/2+1);
        }
        else ans+=n/2*qpow(3,n/2)+n/2*qpow(3,n/2+1);
        ans/=2*n;
        printf("%lld\n",ans);
    }
}
int main(){work();}
```

E.

题意:

有n个球，给出每个小球是什么颜色，同颜色的小球间没有区别，问选出k个球有多少种不同的方案

题解:

多重集组合数，不满足$\forall i\ cnt_i \geq k$，不能直接用公式算，直接dp的话复杂度不能接受，于是用生成函数做

用$cnt[i]$表示第i种颜色有多少个球，那么对于第i种颜色，有生成函数:

$$F_i(x) = 1 + x + x^2 + \cdots + x^{cnt[i]}$$

系数都是1是因为选出这些小球只有1种方案，因为同种颜色之间的球没有区别，高于$cnt[i]$次的系数是0，因为显然没有办法选出超过$cnt[i]$个这种颜色的球

然后把这些多项式乘起来，第k次项就是答案。

然而。最多有n个多项式，这些多项式乘起来最高次项也差不多是n的，那么每一个多项式都做大小为n的FFT的复杂度是$O(n^2 \log n)$的，还是不够快。

由于这n个多项式的总长度是n，那么就是说会有许多小的多项式，而两个小的多项式乘起来比较快，可以考虑每次把两个小的多项式乘起来，合并成一个大一点的多项式，类似霍夫曼编码。

这样的复杂度应该是$O(n\log^2 n)$的。另外模数1009不能NTT，直接用FFT精度是够的

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;typedef double db;
typedef pair<int, int> pii;typedef pair<ll, ll> pll;
typedef pair<int,ll> pil;typedef pair<ll,int> pli;
#define Fi first
#define Se second
#define _Out(a) cerr<<#a<<" = "<<(a)<<endl
const int INF = 0x3f3f3f3f, MAXN = 1e6 + 50;
const ll LINF = 0x3f3f3f3f3f3f3f3f, MOD = 1009;
const db Pi = acos(-1), EPS = 1e-1;
void test(){cerr << "\n";}template<typename T,typename...Args>void test(T
x,Args...args){cerr<<x<<" ";test(args...);}
inline ll qpow(ll a, ll b){return b?((b&1)?
a*qpow(a*a%MOD,b>>1)%MOD:qpow(a*a%MOD,b>>1))%MOD:1;}
inline ll qpow(ll a, ll b,ll c){return b?((b&1)?
a*qpow(a*a%c,b>>1,c)%c:qpow(a*a%c,b>>1,c)) %c:1;}
inline ll gcd(ll a,ll b){return b?gcd(b,a%b):a;}
inline ll cede(ll a,ll b){if(b<0)return cede(-a,-b);if(a<0)return a/b;return
(a+b-1)/b;}
inline ll flde(ll a,ll b){if(b<0)return flde(-a,-b);if(a<0)return (a-
b+1)/b;return a/b;}
inline int sign(db x){return x<-EPS ? -1:x>EPS;}
inline int dbcmp(db l,db r){return sign(l - r);}
namespace Fast_IO{ //orz laofu
    const int MAXL((1 << 18) + 1);int iof, iotp;
    char ioif[MAXL], *ioiS, *ioiT, ioof[MAXL],*iooS=ioof,*iooT=ioof+MAXL-
1,ioc,iost[55];
    char Getchar(){
        if (ioiS == ioiT){
            ioiS=ioif;ioiT=ioiS+fread(ioif,1,MAXL,stdin);return (ioiS == ioiT ?
EOF : *ioiS++);
        }else return (*ioiS++);
    }
    void Write(){fwrite(ioof,1,iooS-ioof,stdout);iooS=ioof;}
    void Putchar(char x){*iooS++ = x;if (iooS == iooT)Write();}
    inline int read(){
        int x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return
x*iof;
    }
    inline long long read_ll(){
        long long x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return
x*iof;
    }
    void Getstr(char *s, int &l){
        for(ioc=Getchar();ioc==' '||ioc=='\n'||ioc=='\t';)ioc=Getchar();
```

```
        if(ioc==EOF)Write(),exit(0);
        for(l=0;!(ioc=='
'||ioc=='\n'||ioc=='\t'||ioc==EOF);ioc=Getchar())s[l++]=ioc;s[l] = 0;
    }
    template <class Int>void Print(Int x, char ch = '\0'){
        if(!x)Putchar('0');if(x<0)Putchar('-'),x=-
x;while(x)iost[++iotp]=x%10+'0',x/=10;
        while(iotp)Putchar(iost[iotp--]);if (ch)Putchar(ch);
    }
    void Putstr(const char *s){for(int i=0,n=strlen(s);i<n;++i)Putchar(s[i]);}
} // namespace Fast_IO
using namespace Fast_IO;
namespace DXS
{
    struct CP
    {
        long double x,y;
        CP (long double xx=0,long double yy=0){x=xx;y=yy;}
        CP operator +(const CP &b){return CP(x+b.x,y+b.y);}
        CP operator -(const CP &b){return CP(x-b.x,y-b.y);}
        CP operator *(const CP &b){return CP(x*b.x-y*b.y,x*b.y+y*b.x);}
    };

    const int G = 3, Gi = 332748118;
    int bit=0,lim=1;
    int rev[MAXN];
    int init(int N){//初始长度
        lim=1,bit=0;while(lim<N)lim<<=1,bit++;
        for(int i=0;i<lim;i++)rev[i]=(rev[i>>1]>>1)|((i&1)<<(bit-1));
        return lim;
    }
    void NTT(int *A,int lim,int inv){
        for(int i=0;i<lim;i++)if(i<rev[i])swap(A[i],A[rev[i]]);
        for(int mid=1;mid<lim;mid<<=1){
            int W=qpow(inv==1?G:Gi,(MOD-1)/(mid<<1));
            for(int j=0;j<lim;j+=(mid<<1)){
                int w=1;for(int k=0;k<mid;k++,w=1ll*w*W%MOD){
                    int x=A[j+k],y=1ll*w*A[j+k+mid]%MOD;
                    A[j+k]=(x+y)%MOD;A[j+k+mid]=(x-y+MOD)%MOD;
                }
            }
        }
        if(inv==-1)for(int i=0,iv=qpow(lim,MOD-
2);i<lim;i++)A[i]=1ll*iv*A[i]%MOD;
    }
    void FFT(CP *A,int lim,int inv){
        for(int i=0;i<lim;i++)if(i<rev[i])swap(A[i],A[rev[i]]);
        for(int mid=1;mid<lim;mid<<=1){
            CP XX(cos(Pi/mid),inv*sin(Pi/mid));
            for(int j=0;j<lim;j+=(mid<<1)){
                CP w(1,0);for(int k=0;k<mid;k++,w=w*XX){
                    CP x=A[j+k],y=w*A[j+mid+k];
                    A[j+k]=x+y;A[j+mid+k]=x-y;
                }
            }
        }
        if(inv==-1)for(int i=0;i<lim;i++)A[i].x/=lim;
    }
```

```
    void Getdev(int *A,int *B,int n){for(int i=0;i<n-1;++i)B[i]=1ll*A[i+1]*
(i+1)%MOD;}
    void Getint(int *A,int *B,int n){for(int i=n;i>=1;--i)B[i]=1ll*A[i-
1]*qpow(i,MOD-2)%MOD;B[0]=0;}
    void Rightshift(int *A,int *B,int n,int k){for(int i=n-1;i>=k;--i)B[i]=A[i-
k];for(int i=0;i<min(k,n);i++)B[i]=0;}
    void Leftshift(int *A,int *B,int n,int k){for(int i=n-k-1;i>=0;--
i)B[i]=A[i+k];for(int i=n-1;i>=max(0,n-k);i--)B[i]=0;}
    void mulC(int *A,int n,int c){for(int i=0;i<n;i++)A[i]=1ll*A[i]*c%MOD;}
    void mul(int *A,int *B,int *C,int n,int m){//NOTE: A[i] and B[i] must be 0
when i>n, i>m (until 2n,2m)
        int len=init(n+m);NTT(A,len,1);if(A!=B)NTT(B,len,1);
        for(int i=0;i<len;i++)C[i]=1ll*A[i]*B[i]%MOD;NTT(C,len,-1);
    }
    //int tmpA[MAXN];
    //void Getinv(int *A,int *B,int n)// return B;
    //{
    //  int now=1,len=2;
    //  for(int i=0;i<=n+n;i++)B[i]=0;
    //  B[0]=qpow(A[0],MOD-2);
    //  while(now<n+n){
    //      for(int i=0;i<now;i++)tmpA[i]=A[i];
    //      for(int i=now;i<len;i++)tmpA[i]=0;
    //      init(len);NTT(tmpA,len,1);NTT(B,len,1);
    //      for(int i=0;i<len;i++)B[i]=1ll*B[i]*(2-
1ll*tmpA[i]*B[i]%MOD+MOD)%MOD;
    //      NTT(B,len,-1);for(int i=now;i<len;i++)B[i]=0;
    //      now<<=1;len<<=1;
    //  }
    //}
    //int invA[MAXN],devA[MAXN];
    //void Getln(int *A,int *B,int n){
    //  for(int i=0;i<n+n;i++)invA[i]=devA[i]=0;
    //  Getinv(A,invA,n);Getdev(A,devA,n);
    //  mul(invA,devA,invA,n,n);Getint(invA,B,n);
    //  B[n]=0;
    //}
    //int lnB[MAXN];
    //void Getexp(int *A,int *B,int n)
    //{
    //  int now=1,len=2;
    //  for(int i=0;i<=n+n;i++)B[i]=0;
    //  B[0]=1;
    //  while(now<n+n)
    //  {
    //      for(int i=0;i<len;i++)lnB[i]=0;
    //      Getln(B,lnB,now);
    //      for(int i=0;i<now;i++)lnB[i]=(MOD-lnB[i]+A[i]+(i==0))%MOD;
    //      init(len);NTT(B,len,1);NTT(lnB,len,1);
    //      for(int i=0;i<len;i++)B[i]=1ll*lnB[i]*B[i]%MOD;
    //      NTT(B,len,-1);for(int i=now;i<len;i++)B[i]=0;
    //      now<<=1;len<<=1;
    //  }
    //}
    //int tmpA2[MAXN],lnA[MAXN];
    //void Getpow(int *A,int *B,int n,int k1,int k2,int k3)//Note: only when
a[0] = 1 can be used for sqrt, else find ecsy for a[0]
    //{
```

```
    //  for(int i=0;i<n+n;i++)tmpA[i]=0;
    //  int cnt0=0;while(A[cnt0]==0&&cnt0<n)cnt0++;
    //  int xs=A[cnt0],invxs=qpow(xs,MOD-2);
    //  xs=qpow(xs,k2);
    //  Leftshift(A,tmpA2,n,cnt0);
    //  mulC(tmpA2,n,invxs);
    //  Getln(tmpA2,lnA,n);
    //  mulC(lnA,n,k1);
    //  Getexp(lnA,B,n);
    //  Rightshift(B,B,n,min((ll)INF,1ll*cnt0*k3));
    //  mulC(B,n,xs);
    //}

    /*void GetSqrt(int *A,int *B,int n)
    {
    int k=qpow(2,MOD-2);
    for(int i=0;i<n+n;i++)tmpA[i]=0;
    int cnt0=0;while(A[cnt0]==0&&cnt0<n)cnt0++;
    int xs=A[cnt0],invxs=qpow(xs,MOD-2);
    xs=ECSY::solve(xs,MOD);
    xs=min(xs,(int)MOD-xs);
    Leftshift(A,tmpA2,n,cnt0);
    mulC(tmpA2,n,invxs);
    Getln(tmpA2,lnA,n);
    mulC(lnA,n,k);
    Getexp(lnA,B,n);
    Rightshift(B,B,n,min((ll)INF,1ll*cnt0/2));
    mulC(B,n,xs);
    }*/
}
int cnt[MAXN];
struct PNode
{
    int val,id;
    bool operator<(const PNode &R)const
    {
        return val>R.val;
    }
};
priority_queue<PNode>pq;
vector<int>ve[MAXN];
DXS::CP a[MAXN],b[MAXN];
int toint(long double x,int M)
{
    ll k=(x+EPS)/M;
    x-=k*M;
    return x+2*EPS;
}
void merge(int id1,int id2)
{
    int maxv=ve[id1].size()+ve[id2].size()-1;
    for(int i=0;i<4*maxv+2;i++)a[i]=b[i]={0,0};
    for(int i=0;i<ve[id1].size();i++)a[i].x=ve[id1][i];
    for(int i=0;i<ve[id2].size();i++)b[i].x=ve[id2][i];
    int len = DXS::init(maxv+1);
    DXS::FFT(a,len,1);DXS::FFT(b,len,1);
    for(int i=0;i<len;i++)a[i]=a[i]*b[i];
    DXS::FFT(a,len,-1);
```

```
    ve[id1].clear();ve[id2].clear();
    for(int i=0;i<maxv;i++)ve[id1].push_back(toint(a[i].x,(int)MOD));
}
void work()
{
    int n=read(),m=read(),k=read();
    for(int i=1;i<=n;i++)
    {
        int x=read();
        cnt[x]++;
    }
    int cntS=0;
    for(int i=1;i<=m;i++)
    {
        if(!cnt[i])continue;
        ++cntS;
        for(int j=0;j<=cnt[i];j++)ve[cntS].push_back(1);
        pq.push({(int)ve[cntS].size(),cntS});
    }
    while(pq.size()>1)
    {
        PNode tmp1=pq.top();pq.pop();
        PNode tmp2=pq.top();pq.pop();

        merge(tmp1.id,tmp2.id);
        pq.push({(int)ve[tmp1.id].size(),tmp1.id});

    }
    printf("%d\n",ve[pq.top().id][k]);
}
int main(){
        work();
}
```

F.

题意:

忘得差不多了

题解:

当时比赛的时候，我（还有大部分队伍）推了很久的dp式子，然后赛后发现dls直接BM秒了。

至于怎么求前几项，一般可以暴力，也可以手算，也可以写个正解。。

（板子网上随便找的，可能不是很靠谱， 段老师应该也留下了一些BM板子，可以自己找找）

```
#include <cstdio>
#include <cstring>
#include <cmath>
#include <algorithm>
#include <vector>
#include <string>
#include <set>
```

```
#include <cassert>
using namespace std;
#define rep(i,a,n) for (ll i=a;i<n;i++)
#define per(i,a,n) for (ll i=n-1;i>=a;i--)
#define pb push_back
#define mp make_pair
#define all(x) (x).begin(),(x).end()
#define fi first
#define se second
#define SZ(x) ((ll)(x).size())
typedef long long ll;
typedef vector<ll> VI;
typedef pair<ll, ll> PII;
const ll mod = 998244353ll;
ll powmod(ll a, ll b) { ll res = 1; a %= mod; assert(b >= 0); for (; b; b >>= 1)
{ if (b & 1)res = res*a%mod; a = a*a%mod; }return res; }
ll _, n;
namespace linear_seq {
    const ll N = 10010;
    ll res[N], base[N], _c[N], _md[N];
    vector<ll> Md;
    void mul(ll *a, ll *b, ll k) {
        rep(i, 0, k + k) _c[i] = 0;
        rep(i, 0, k) if (a[i]) rep(j, 0, k) _c[i + j] = (_c[i + j] + a[i] *
b[j]) % mod;
        for (ll i = k + k - 1; i >= k; i--) if (_c[i])
            rep(j, 0, SZ(Md)) _c[i - k + Md[j]] = (_c[i - k + Md[j]] - _c[i] *
_md[Md[j]]) % mod;
        rep(i, 0, k) a[i] = _c[i];
    }
    ll solve(ll n, VI a, VI b) {
        ll ans = 0, pnt = 0;
        ll k = SZ(a);
        assert(SZ(a) == SZ(b));
        rep(i, 0, k) _md[k - 1 - i] = -a[i]; _md[k] = 1;
        Md.clear();
        rep(i, 0, k) if (_md[i] != 0) Md.push_back(i);
        rep(i, 0, k) res[i] = base[i] = 0;
        res[0] = 1;
        while ((1ll << pnt) <= n) pnt++;
        for (ll p = pnt; p >= 0; p--) {
            mul(res, res, k);
            if ((n >> p) & 1) {
                for (ll i = k - 1; i >= 0; i--) res[i + 1] = res[i]; res[0] = 0;
                rep(j, 0, SZ(Md)) res[Md[j]] = (res[Md[j]] - res[k] *
_md[Md[j]]) % mod;
            }
        }
        rep(i, 0, k) ans = (ans + res[i] * b[i]) % mod;
        if (ans < 0) ans += mod;
        return ans;
    }
    VI BM(VI s) {
        VI C(1, 1), B(1, 1);
        ll L = 0, m = 1, b = 1;
        rep(n, 0, SZ(s)) {
            ll d = 0;
            rep(i, 0, L + 1) d = (d + (ll)C[i] * s[n - i]) % mod;
```

```
            if (d == 0) ++m;
            else if (2 * L <= n) {
                VI T = C;
                ll c = mod - d*powmod(b, mod - 2) % mod;
                while (SZ(C) < SZ(B) + m) C.pb(0);
                rep(i, 0, SZ(B)) C[i + m] = (C[i + m] + c*B[i]) % mod;
                L = n + 1 - L; B = T; b = d; m = 1;
            }
            else {
                ll c = mod - d*powmod(b, mod - 2) % mod;
                while (SZ(C) < SZ(B) + m) C.pb(0);
                rep(i, 0, SZ(B)) C[i + m] = (C[i + m] + c*B[i]) % mod;
                ++m;
            }
        }
        return C;
    }
    ll gao(VI a, ll n) {
        VI c = BM(a);
        c.erase(c.begin());
        rep(i, 0, SZ(c)) c[i] = (mod - c[i]) % mod;
        return solve(n, c, VI(a.begin(), a.begin() + SZ(c)));
    }
};
int main() {
    int T;
    scanf("%d",&T);
    while(T--)
    {
        int n;
        scanf("%d",&n);
        VI vec={45,4950,500400,50103000,19816235,992019147,476883753,957046356};
        printf("%I64d\n", linear_seq::gao(vec, n-1));
    }
}
```

G.

题意:

给一排n个格子染色，总共有m种颜色可以染，只有当相邻两个格子颜色不同才是合法的，求恰好用了k种不同颜色的染色方案数

题解:

假设没有恰好k个不同颜色的限制，那么答案很显然就是$m \cdot (m-1)^{n-1}$

首先注意到m其实没什么用，只是有C_m^k种选颜色的方案数，所以我们假设只有k种颜色，也就是求所有k种颜色都要用的方案数，最后乘C_m^k即可。

令$F(k)$等于有k种颜色，但不要每种颜色都出现的染色方案数，显然$F(k) = k \cdot (k-1)^{n-1}$

令$G(k)$等于有k种颜色，且每种颜色都要用到的方案数。于是:

$$F(k) = \sum_{i=1}^{k} \binom{k}{i} G(i)$$

就是枚举实际上用了i种颜色，那么就有$\binom{k}{i}$种颜色的选择，然后对于每一种颜色选择，有$G(i)$种每个颜色都出现的方案数。

然后就是套用二项式反演的公式:

$$G(k)=\sum_{i=1}^{k}(-1)^{k-i}\binom{k}{i}F(i)$$

然后$G(k)\cdot\binom{m}{k}$就是答案，当然m很大需要用递推来求组合数

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;typedef double db;
typedef pair<int, int> pii;typedef pair<ll, ll> pll;
typedef pair<int,ll> pil;typedef pair<ll,int> pli;
#define Fi first
#define Se second
#define _Out(a) cerr<<#a<<" = "<<(a)<<endl
const int INF = 0x3f3f3f3f, MAXN = 2e6 + 50;
const ll LINF = 0x3f3f3f3f3f3f3f3f, MOD = 1e9+7;
const db Pi = acos(-1), EPS = 1e-5;
void test(){cerr << "\n";}template<typename T,typename...Args>void test(T
x,Args...args){cerr<<x<<" ";test(args...);}
inline ll qpow(ll a, ll b){return b?((b&1)?
a*qpow(a*a%MOD,b>>1)%MOD:qpow(a*a%MOD,b>>1))%MOD:1;}
inline ll qpow(ll a, ll b,ll c){return b?((b&1)?
a*qpow(a*a%c,b>>1,c)%c:qpow(a*a%c,b>>1,c)) %c:1;}
inline ll gcd(ll a,ll b){return b?gcd(b,a%b):a;}
inline ll cede(ll a,ll b){if(b<0)return cede(-a,-b);if(a<0)return a/b;return
(a+b-1)/b;}
inline ll flde(ll a,ll b){if(b<0)return flde(-a,-b);if(a<0)return (a-
b+1)/b;return a/b;}
inline int sign(db x){return x<-EPS ? -1:x>EPS;}
inline int dbcmp(db l,db r){return sign(l - r);}
namespace Fast_IO{ //orz laofu
    const int MAXL((1 << 18) + 1);int iof, iotp;
    char ioif[MAXL], *ioiS, *ioiT, ioof[MAXL],*iooS=ioof,*iooT=ioof+MAXL-
1,ioc,iost[55];
    char Getchar(){
        if (ioiS == ioiT){
            ioiS=ioif;ioiT=ioiS+fread(ioif,1,MAXL,stdin);return (ioiS == ioiT ?
EOF : *ioiS++);
        }else return (*ioiS++);
    }
    void Write(){fwrite(ioof,1,iooS-ioof,stdout);iooS=ioof;}
    void Putchar(char x){*iooS++ = x;if (iooS == iooT)Write();}
    inline int read(){
        int x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return
x*iof;
    }
    inline long long read_ll(){
        long long x=0;for(iof=1,ioc=Getchar();
(ioc<'0'||ioc>'9')&&ioc!=EOF;)iof=ioc=='-'?-1:1,ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
```

```
        for(x=0;ioc<='9'&&ioc>='0';ioc=Getchar())x=(x<<3)+(x<<1)+(ioc^48);return
x*iof;
    }
    void Getstr(char *s, int &l){
        for(ioc=Getchar();ioc==' '||ioc=='\n'||ioc=='\t';)ioc=Getchar();
        if(ioc==EOF)Write(),exit(0);
        for(l=0;!(ioc=='
'||ioc=='\n'||ioc=='\t'||ioc==EOF);ioc=Getchar())s[l++]=ioc;s[l] = 0;
    }
    template <class Int>void Print(Int x, char ch = '\0'){
        if(!x)Putchar('0');if(x<0)Putchar('-'),x=-
x;while(x)iost[++iotp]=x%10+'0',x/=10;
        while(iotp)Putchar(iost[iotp--]);if (ch)Putchar(ch);
    }
    void Putstr(const char *s){for(int i=0,n=strlen(s);i<n;++i)Putchar(s[i]);}
} // namespace Fast_IO
using namespace Fast_IO;

int n,m,k;
ll F(ll m,ll n)
{
    if(m==0)return 0;
    return 1ll*m*qpow(m-1,n-1)%MOD;
}
int fac[MAXN],invfac[MAXN],inv[MAXN];
void init()
{
    fac[0]=1;for(int i=1;i<MAXN;i++)fac[i]=1ll*fac[i-1]*i%MOD;
    invfac[MAXN-1]=qpow(fac[MAXN-1],MOD-2);
    for(int i=MAXN-2;i>=0;--i)invfac[i]=1ll*invfac[i+1]*(i+1)%MOD;
    for(int i=1;i<MAXN-1;i++)inv[i]=1ll*invfac[i]*fac[i-1]%MOD;
}
int C(int n,int m){
    if(n==0&&m==n)return 1;
    if(m<0||m>n)return 0;
    return 1ll*fac[n]*invfac[m]%MOD*invfac[n-m]%MOD;
}
ll CC(int n,int m){
    ll ret=1;
    for(int i=1;i<=m;i++)
        ret=1ll*ret*(n-i+1)%MOD*inv[i]%MOD;
    return ret;
}
void work()
{
    n=read(),m=read(),k=read();
    if(k>m)
    {
        printf("0\n");
        return;
    }
    ll ans=0,f=1;
    for(int i=k;i>=0;--i)
    {
        ans=(ans+f*C(k,i)%MOD*F(i,n)%MOD+MOD)%MOD;
        f*=-1;
    }
    ans=(ans*CC(m,k)+MOD)%MOD;
```

```
    printf("%lld\n",ans);
}
int main()
{
    init();
    int T=read();for(int cas=1;cas<=T;cas++)
        printf("Case #%d: ",cas),
        work();
}
```

H.

题意:

求贝尔数

题解:

用递推公式$O(N^2)$求即可

I.

题意:

求自然数幂和$\sum_{i=1}^{n} i^k$

题解:

可以用三种方法求，这里用的是伯努利数。好像拉格朗日插值会被卡掉

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;typedef double db;
typedef pair<int, int> pii;typedef pair<ll, ll> pll;
typedef pair<int,ll> pil;typedef pair<ll,int> pli;
#define Fi first
#define Se second
#define _Out(a) cerr<<#a<<" = "<<(a)<<endl
const int INF = 0x3f3f3f3f, MAXN = 2e6 + 50;
const ll LINF = 0x3f3f3f3f3f3f3f3f, MOD = 1e9+7;
const db Pi = acos(-1), EPS = 1e-5;
void test(){cerr << "\n";}template<typename T,typename...Args>void test(T
x,Args...args){cerr<<x<<" ";test(args...);}
inline ll qpow(ll a, ll b){return b?((b&1)?
a*qpow(a*a%MOD,b>>1)%MOD:qpow(a*a%MOD,b>>1))%MOD:1;}
inline ll qpow(ll a, ll b,ll c){return b?((b&1)?
a*qpow(a*a%c,b>>1,c)%c:qpow(a*a%c,b>>1,c)) %c:1;}
ll B[MAXN];
int fac[MAXN],invfac[MAXN],inv[MAXN];
int C(int n,int m){
    if(n==0&&m==n)return 1;
    if(m<0||m>n)return 0;
    return 1ll*fac[n]*invfac[m]%MOD*invfac[n-m]%MOD;
}
```

```
void init()
{
    fac[0]=1;for(int i=1;i<MAXN;i++)fac[i]=1ll*fac[i-1]*i%MOD;
    invfac[MAXN-1]=qpow(fac[MAXN-1],MOD-2);
    for(int i=MAXN-2;i>=0;--i)invfac[i]=1ll*invfac[i+1]*(i+1)%MOD;
    for(int i=1;i<MAXN-1;i++)inv[i]=1ll*invfac[i]*fac[i-1]%MOD;

    B[0]=1;
    for(int i=1;i<2005;i++)
    {
        for(int j=0;j<i;j++)B[i]=(B[i]+1ll*C(i+1,j)*B[j]%MOD)%MOD;
        B[i]=-1ll*B[i]*inv[i+1]%MOD;
        B[i]=(B[i]+MOD)%MOD;
    }
}

void work()
{
    ll n,k;
    scanf("%lld%lld",&n,&k);n++;
    ll ans=0;
    for(int i=0;i<=k;i++)
    {
        ans=(ans+C(k+1,i)*B[i]%MOD*qpow(n,k+1-i)%MOD+MOD)%MOD;
    }
    ans=ans*inv[k+1]%MOD;
    printf("%lld\n",ans);
}
int main()
{
    init();
    int T;scanf("%d",&T);for(int i=1;i<=T;i++)
        work();
}
```

J.

暴力前几项，然后拉格朗日插值即可