题意：X\*Y\*Z。每个位置(x,y,z)有一盏灯，初始时是灭的。K轮操作。每轮任意选两盏灯(可以相同)将其之间的所有灯翻转。问最后明着的灯的数量期望。

思路：先求出每盏灯在一轮中被改变的概率，对于(i,j,k)位置的，只要选择的两盏灯不在同一侧就行。那么可以对于每一维分别算，比如对于X这一维，就是减去在其同侧的。然后计算K轮被改变奇数次的概率就是该盏灯最后对答案的贡献。

设f(x)为选择x次区间时，状态改变次数为奇数的概率，g(x)为状态改变次数为偶数的概率。  
f(x)=(1-p)\*f(x-1)+p\*g(x-1)  
g(x)=(1-p)\*g(x-1)+p\*f(x-1)  
f(x)+g(x)=1  
因此f(x)=(1-2p)\*f(x-1)+p,两边同时除以(1-2p)^x，再化简成等比数列通项公式f(x)=0.5-0.5\*(1-2p)^x

f(x)=(1-2p)\*f(x-1)+p

同除(1-2p)^x得

令an=

则 an-an-1=

所以

an-an-1=

an-1 - an-2=

an-2 - an-3=

······

a1-a0=

两边相加得

an-a0=p()

由定义得a0=0

等式右边用等比数列求和可得

an=p

可得a(n)=0.5-0.5\*(1-2p)^n

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

const int INF=1000000007;

double cal(int i,int x)

{

return x\*x-(i-1)\*(i-1)-(x-i)\*(x-i);

}

int main()

{

// freopen("input.txt","r",stdin);

// ios::sync\_with\_stdio(false);

// cin.tie(0);

int T,x,y,z,k;

cin>>T;

for(int cas=1;cas<=T;cas++)

{

cin>>x>>y>>z>>k;

double t=1.0\*x\*x\*y\*y\*z\*z;

double t=x\*x\*y\*y\*z\*z;

double p;

double ans(0);

for(int i=1;i<=x;i++)

for(int j=1;j<=y;j++)

for(int h=1;h<=z;h++)

{

p=cal(i,x)\*cal(j,y)\*cal(h,z)/t;

ans+=0.5-0.5\*pow(1.0-2.0\*p,(double)k);

}

cout<<"Case "<<cas<<": "<<fixed<<setprecision(7)<<ans<<endl;

}

return 0;

}