```
#include<bits/stdc++.h>
using namespace std;

#define mx 10000
#define ll long long
#define lf double
#define SQ(a) a * a
#define inf 0x3f3f3f3f

#define sfll(a) scanf("%lld",&a)
#define pfll(a) printf("%lld",a)
#define sflf(a) scanf("%lf",&a)
#define pflf(a) printf("%lf",a)
#define sff(a) scanf("%f",&a)
#define pff(a) printf("%f",a)
#define sf(a) scanf("%d",&a)
#define pf(a) printf("%d",a)
#define line  puts("")
#define space printf(" ")

#define fread freopen("input.txt","rt",stdin)
#define fwrite freopen("output.txt","wt",stdout)
//  int fx[]={+1,-1,+0,+0};
//  int fy[]={+0,+0,+1,-1};
//  int fx[]={+0,+0,+1,-1,-1,+1,-1,+1};
//  int fy[]={-1,+1,+0,+0,+1,+1,-1,-1};
//  int fx[]={+0,+0,+1,-1,-1,+1,-1,+1};   // Kings Move
//  int fy[]={-1,+1,+0,+0,+1,+1,-1,-1};  // Kings Move
//  int fx[]={-2, -2, -1, -1,  1,  1,  2,  2};  // Knights Move
//  int fy[]={-1,  1, -2,  2, -2,  2, -1,  1}; // Knights Move

///sieve
/*
vector<ll>prime;
bool chk[mx+5];
void sieve()
{
    prime.push_back(2);
    for(ll i=4; i<=mx; i+=2)
        chk[i] = true;
    for(ll i=3; i<=mx; i+=2)
    {
        if(chk[i] == false)
        {
            prime.push_back(i);
            for(ll j= i * i ; j <=mx ; j+=( 2 *  i))
                chk[j] = true;
        }
    }

}
*/

///prime_factor
/*

ll fact[mx+ 5];
void p_fact()
{
    for(ll i = 2; i<=mx; i++)
        fact[i] = i;
    for(ll i = 2; i<=mx; i+=2)
        fact[i] = 2;
    for(ll i = 3; i<=mx; i+=2)
    {
        if(fact[i] == i)
            for(ll j= i * i; j<=mx ; j+=(2 * i))
                fact[j] = i;
    }

}
*/
///bigmod

/*

template<typename T>inline T bigmod(T a, T b, T m)
{
    if (b == 0)return 1;
    else if (b % 2 == 0)return SQ(bigmod(a, b / 2, m)) % m;
    else return (a % m*bigmod(a, b - 1, m)) % m;
}
*/

///bigmod_inverse
/*
template < class T > T modInverse(T b, T m)
{
    return BIGMOD(b, m-2, m);
}
*/

///Fast expo
/*

ll power(ll b, ll p)
```

```
{
    ll ans = 1;
    while(p)
    {
        if(p&1)ans*=b;
        b*=b;
        p/=2;
    }
    return ans;
}
*/
///distance between two point
/*
lf dist(int x,int y,int x1,int y1)
{
    return sqrt(((x1-x)*(x1-x))+((y1-y)*(y1-y)));
}
*/

int main()
{


}
```