// 高斯消元

//long double!

const long double eps=1e-10; //if failed, try 1e-2

inline long double myabs(LL x)

{

return x>=0?x:-x;

}

inline void Swap(int r1,int r2)

{

int i;

for (i=1;i<=n+1;i++) swap(b[r1][i],b[r2][i]);

}

inline void Remove(int r1,int r2,int col)

{

if (b[r2][col]==0) return;

long double tmp=b[r2][col];

for (int i=col;i<=n+1;i++) b[r2][i]-=b[r1][i]\*tmp;

}

bool solve()

{

int i,j,rr;

//消元

//注意不要用long long求lcm而是用long double，不然会爆long long

for (j=1;j<=n-1;j++)

{

rr=j;

while (rr<=n && myabs(b[rr][j])<eps) rr++;

if (rr>n) return false; //有自由元，说明有多解

if (j!=rr) Swap(j,rr);

if (myabs(b[j][j]-1.0)>eps)

{

long double tmp=b[j][j];

for (i=1;i<=n+1;i++) b[j][i]/=tmp;

}

for (i=rr+1;i<=n;i++) Remove(j,i,j);

}

if (myabs(b[n][n])<eps) return false;

sol[n]=b[n][n+1]/b[n][n];

//回代

long double res;

for (i=n-1;i>=1;i--)

{

res=b[i][n+1];

for (j=i+1;j<=n;j++) res-=sol[j]\*b[i][j];

if (myabs(b[i][i]-1)>eps) return false;

sol[i]=res/b[i][i];

}

return true;

}

// 求方程解的个数

//long double!

const long double eps=1e-10; //if failed, try 1e-2

bool taken[148];

//值为true表示这是一个有效方程

int t,n;

//n未知数个数,t方程个数

inline long double myabs(LL x)

{

return x>=0?x:-x;

}

inline void Remove(int r1,int r2,int col)

{

double tmp=b[r1][col];

for (int i=1;i<=n+1;i++) b[r1][i]-=b[r2][i]\*tmp;

}

void solve()

{

int i,j,rr;

//消元

//注意不要用long long求lcm而是用double，不然会爆long long

for (j=1;j<=n;j++)

{

rr=1

while (rr<=t && (taken[rr] || myabs(b[rr][j])<eps)) rr++;

if (rr>t) continue; //有自由元，说明有多解

taken[rr]=true;

if (myabs(b[rr][j]-1.0)>eps)

{

long double tmp=b[rr][j];

for (i=1;i<=n+1;i++) b[rr][i]/=tmp;

}

for (i=1;i<=t;i++) if (!taken[i] && myabs(b[i][j])>eps) Remove(i,rr,j);

}

}

int main ()

{

//init

solve();

int i,j,Free=n;

for (i=1;i<=t;i++) if (taken[i]) Free--;

//自由元个数=Free

}