伪代码（Ref：<http://www.cnblogs.com/heaad/archive/2010/12/20/1911614.html> ）

/\*

\* J(y)：在状态y时的评价函数值

\* Y(i)：表示当前状态

\* Y(i+1)：表示新的状态

\* r： 用于控制降温的快慢

\* T： 系统的温度，系统初始应该要处于一个高温的状态

\* T\_min ：温度的下限，若温度T达到T\_min，则停止搜索

\*/

while( T > T\_min )

{

　　dE = J( Y(i+1) ) - J( Y(i) ) ;

　　if ( dE >=0 ) //表达移动后得到更优解，则总是接受移动

Y(i+1) = Y(i) ; //接受从Y(i)到Y(i+1)的移动

　　else

　　{

// 函数exp( dE/T )的取值范围是(0,1) ，dE/T越大，则exp( dE/T )也

if ( exp( dE/T ) > random( 0 , 1 ) )

Y(i+1) = Y(i) ; //接受从Y(i)到Y(i+1)的移动

　　}

　　T = r \* T ; //降温退火 ，0<r<1 。r越大，降温越慢；r越小，降温越快

　　/\*

　　\* 若r过大，则搜索到全局最优解的可能会较高，但搜索的过程也就较长。若r过小，则搜索的过程会很快，但最终可能会达到一个局部最优值

　　\*/

　　i ++ ;

}

**Ver 1.0：【hdu5017】**

**求dis(x,y,z)函数的最小值（x、y、z满足z=calc(x,y)）**

int dx[10]={-1, 0, 1,1,1,0,-1,-1};

int dy[10]={-1,-1,-1,0,1,1, 1, 0}; //点(x,y)向周围8个方向扩散

#define Mod 1000000007

#define eps 1e-8

double sa() //Simulated\_Annealing

{

double x = 0,y = 0,z = sqrt(1.0/C); //当前最优解

double step = 1.0, rate = 0.99; //step：系统温度 rate：控制降温的速度

while(step > eps)

{

for(int k=0;k<8;k++) //(x,y)可以向8个方向扩散，z直接用解析式算出，这样就保证了(x,y,z)一定在椭球面上

{

double kx = x + step\*dx[k];

double ky = y + step\*dy[k];

double kz = calc(kx,ky);

if(kz >= Mod) continue; //不符合要求的筛掉

if(dis(kx,ky,kz) < dis(x,y,z))

{

x = kx,y = ky,z = kz;

}

}

step \*= rate;

}

return dis(x,y,z);

}

**Ver2.0：【hdu3932】**

**求使得函数dis(x,y)值最小的点**

#include<cmath>

#include<ctime>

#define eps 1e-3

#define pi acos(-1.0)

#define POI 15 //独立跑POI次，找其中最小的 tp[1..POI]是随机的初值

#define RUN 40 //迭代次数，本题中即点(tx,ty)向RUN个方向发散

void sa()

{

for(int i=1;i<=POI;i++) //赋随机的初值

{

tp[i].x=(rand()%1000+1)/1000.0\*X;

tp[i].y=(rand()%1000+1)/1000.0\*Y;

sol[i]=dis(tp[i].x,tp[i].y);

//printf("%.1f~%.1f=%.1f\n",tp[i].x,tp[i].y,sol[i]);

}

double step=1.0\*max(X,Y)/sqrt(1.0\*N);

while(step>eps) // Simulated Annealing

{

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

{

double kx=tp[i].x,ky=tp[i].y;

double tx=kx,ty=ky;

for(int j=0;j<RUN;j++)

{

double angle=(rand()%1000+1)/1000.0\*10\*pi;

kx=tx+cos(angle)\*step;

ky=ty+sin(angle)\*step;

if((kx>X)||(ky>Y)||(kx<0)||(ky<0)) continue;

double tmp=dis(kx,ky);

if(tmp<sol[i])

{

tp[i].x=kx; tp[i].y=ky;

sol[i]=tmp;

}

}

}

step\*=0.80;

}

ans=(X\*X+Y\*Y)\*100.0; //找答案（最值）

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

{

if(sol[i]<ans)

{

ans=sol[i];

ansi=i;

}

}

//ans是最小值，（tp[ansi].x , tp[ansi].y）是取得最小值的点

}

int main()

{

srand(time(NULL));

while(cin>>X>>Y>>N)

{

sa();

//TODO:Please Code Here

}

return 0;

}