**[A - Wireless Network](https://vjudge.net/problem/POJ-2236" \t "_blank)**

An earthquake takes place in Southeast Asia. The ACM (Asia Cooperated Medical team) have set up a wireless network with the lap computers, but an unexpected aftershock attacked, all computers in the network were all broken. The computers are repaired one by one, and the network gradually began to work again. Because of the hardware restricts, each computer can only directly communicate with the computers that are not farther than d meters from it. But every computer can be regarded as the intermediary of the communication between two other computers, that is to say computer A and computer B can communicate if computer A and computer B can communicate directly or there is a computer C that can communicate with both A and B.   
  
In the process of repairing the network, workers can take two kinds of operations at every moment, repairing a computer, or testing if two computers can communicate. Your job is to answer all the testing operations.

Input

The first line contains two integers N and d (1 <= N <= 1001, 0 <= d <= 20000). Here N is the number of computers, which are numbered from 1 to N, and D is the maximum distance two computers can communicate directly. In the next N lines, each contains two integers xi, yi (0 <= xi, yi <= 10000), which is the coordinate of N computers. From the (N+1)-th line to the end of input, there are operations, which are carried out one by one. Each line contains an operation in one of following two formats:   
1. "O p" (1 <= p <= N), which means repairing computer p.   
2. "S p q" (1 <= p, q <= N), which means testing whether computer p and q can communicate.   
  
The input will not exceed 300000 lines.

Output

For each Testing operation, print "SUCCESS" if the two computers can communicate, or "FAIL" if not.

Sample Input

4 1

0 1

0 2

0 3

0 4

O 1

O 2

O 4

S 1 4

O 3

S 1 4

Sample Output

FAIL

SUCCESS

并查集的简单应用，对每次修好的电脑对其它已经修好的电脑遍历，如果距离小于等于最大通信距离就将他们合并。之后判断2台电脑是不是一个集合中就KO了

#include<iostream>

#include<stdio.h>

#include<cstring>

using namespace std;

struct node

{

int fa,x,y;

};

node a[1010];

int n,d;

short b[1010];

int find(int x)

{

if(x==a[x].fa)

return x;

else

return a[x].fa=find(a[x].fa);

}

void join(node x,node y)

{

int p=find(x.fa);

int q=find(y.fa);

if(q!=p)

if((x.x-y.x)\*(x.x-y.x)+(x.y-y.y)\*(x.y-y.y)<=d\*d)

a[p].fa=q;

}

int main()

{

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

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

char x;

int ok,from,to;

cin>>n>>d;

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

a[i].fa=i;

memset(b,0,sizeof(b));

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

cin>>a[i].x>>a[i].y;

while(cin>>x)

{

if(x=='O')

{

cin>>ok;

b[ok]=1;

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

if(i!=ok&&b[i])

join(a[i],a[ok]);

}

else

{

cin>>from>>to;

if(find(from)==find(to))

cout<<"SUCCESS\n";else cout<<"FAIL\n";

}

}

return 0;

}