Problems

Problem	TRGOVAC	VOKITOKI

Source file	`trgovac.pas`
	`trgovac.c`
	`trgovac.cpp`
		`vokitoki.pas`
		`vokitoki.c`
		`vokitoki.cpp`

Input file	-	`vokitoki.in`
Output file	-	`vokitoki.out`
Time limit per test	5 seconds	5 seconds
Number of tests	10	10
Points per test	9	11
Total points	**90**	**110**
	200	

Mirko has dropped out of college and returned to the good old traveling salesman job he had back when he was sixteen. As is with most traveling salesmen, Mirko has to visit N towns, **each of them exactly once**, advertising his products in each of the towns. Like all businessmen, Mirko uses airplanes to travel between towns.

There is exactly one flight operating between every two towns, the flight is always an hour long and the plane travels between the two towns non stop, without pausing or wasting any time. The departure time is always on a full hour so, for example, if an airplane departs from city A towards city B at noon, he'll land in city B at 1 o'clock, and immediately head back towards city A where he will land at 2 o'clock etc.

Mirko wants to make a tour schedule, so that he spends **exactly one hour** advertising his products in each of the towns on the tour (except for the first and last town) and immediately heads on to the next town on the tour. In other words, he wants to arrange the N towns in an order g_1, …, g_N so that for each i, 1<i<N, Mirko can leave town g_i for town g_i+1 **exactly one hour after** he had arrived to town g_i. Mirko can leave town g_1 **any time he wants** (on a full hour, though).

Mirko doesn't have a flight schedule so he has to call the airline company. Every time he dials up, Mirko picks two different towns A and B, and the operator tells him whether the flight at noon departs from town A towards town B or vice versa. Dialing the airline company is expensive so Mirko has to be smart, not spending too much money doing it.

Additionally, the owner of the airline company may answer the phone. He wants to make a joke with Mirko so he tries to answer Mirko's queries so that Mirko spends as much money dialling as possible. The way he does this is that he rearranges the flight schedule between towns Mirko asks him about.

However, if a clerk answers the phone, he just reads the correct schedule.

The person who answers Mirko's first query will also answer the rest of his queries.

Write a program that will determine a schedule for Mirko (if one exists).

The library

To solve the problem you need to use the library trgovac.o which contains three functions:

`Init` – must be called **exactly once**, at the **start** of your program, without arguments. This function returns the number of towns N. The towns are numbered 1 to N.

```
function Init : integer;
int Init(void);
```

`Pitaj` – this function corresponds to one phone call to the airline company. For a pair of different towns A and B, this function returns A (if the flight at noon departs from town A to town B) or B (if the flight at noon departs from town B to town A). The function will be consistent, i.e. if Pitaj(A,B)=A, then Pitaj(B,A)=A.

```
function Pitaj(a : integer, b : integer) : integer;
int Pitaj(int a, int b);
```

`Gotovo` – call this function when you have the determined the required schedule. It takes an array of integers as its argument, the array being the schedule such that raspored[i] contains the town marked g_i, for 1 ≤ i ≤ N. This function **will terminate** your program.

```
procedure Gotovo(raspored : array of integer);
void Gotovo(int raspored[]);
```

Pascal contestants

You need to include the statement '`uses trgovac`' at the top of your program.

C/C++ contestants

You need to include the statement '`#include "trgovac.h"`' at the top of your program.

There will be **at most 900 towns**, and your program needs to find the solution using **at most 10000 calls** to the function Pitaj.

Testing

For testing purposes, a test library is provided. The library contains only the clerk, and the library has been adjusted to work as follows:

The `trgovac.in` text file contains the answers to all possible queries (calls to the function Pitaj).

The first line contains the number of towns N.

The (i+1)-th line of the file contains the answers to the questions Pitaj(i, j) for each j=i+1, i+2, …, N, separated by spaces.

An example:

```
5
1 1 1 5
3 4 5
3 5
5
```

After your program terminates, i.e upon calling the funcition `Gotovo`, the library will create two files, `trgovac.log`, containing data about the calls to the function Pitaj and `trgovac.out`, containing the number of calls and the schedule, as returned by your program. If your program asks too many questions, the file will contain the number –1, and –2 if a question asked was illegal.

Note: half of the test cases will be answered by the clerk, the other half of them by the lunatic owner ot the company.

VOKITOKI

Mirko and Slavko have finally gotten jobs as locomotive drivers on the Croatian Railroad. As early as the first day of work, they got an assignment. Each of them was to take their locomotive from a certain town and visit as many towns as possible.

Mirko is an experienced driver, so he is afraid of nothing. This was, however, Slavko's first train ride and he can do absolutely nothing by himself. Luckily, all the locomotives have radios, so Slavko can drive the locomotive normally as long as he is within the range of Mirko's radio to give him instructions.

N towns have been given in a coordinate plane. Some of the towns are connected by railroad. Mirko and Slavku start their tour in different towns, and so that they are at most D kilometers away from each other.

The locomotives can use all railroads, at any speed and in any direction. Locomotives can switch railroad tracks in towns only. Mirko and Slavko may be **at most D kilometers** away from each other at any moment in time.

Write a program that will determine **all possible towns** that Slavko can visit, as was described above.

Input data

The first line of the input file contains the numbers N and P, and a real number D, $2 \le N \le 100$, $1 \le P \le 3000$, $1 \le D \le 10{,}000$. The number N is the number of towns, the number P is the number of railroads, and D is the range of the radio in kilometers (a decimal number two digits precise). The towns are numbered 1 to N.

Each of the following N lines contains data describing one town, i.e. two integers, X and Y, $-5000 \le X, Y \le 5000$, representing the town's coordinates.

Each of the following P lines contains data describing one railroad track, i.e. two integers G1 and G2, saying there is a railroad track connecting G1 i G2.

The next line contains the starting positions of Mirko and Slavku, the integers U and V. Mirko starts from town U, Slavko from town V. U and V will represent two towns separated at most D kilometers in distance.

Output data

The output file should contain the numbers of all the towns Slavko can reach. These numbers should be sorted in increasing order, each of them written on a separate line.

Sample test data

vokitoki.in

```
5 4 1.5
0 1
0 0
4 1
4 0
2 2
1 3
1 5
3 5
2 4
2 1
```

vokitoki.out

```
1
3
```

vokitoki.in

```
8 6 4
0 0
4 3
8 0
16 0
0 -1
8 -1
12 -4
16 -1
1 2
2 3
3 4
5 6
6 7
7 8
1 5
```

vokitoki.out

```
5
6
7
8
```

vokitoki.in

```
8 6 2
0 0
1 0
2 0
1 1
0 1
1 3
2 1
1 -10
1 2
2 4
2 3
5 6
6 7
2 8
5 1
```

vokitoki.out

```
1
2
3
4
```