Amusement Park

JOI-kun has a younger sister, IOI-chan. JOI-kun is now playing with IOI-chan in an amusement park called JOIOI park.

There are N attractions in JOIOI park numbered from 0 to $N - 1$. Also, there are M paths in JOIOI park. We can walk each path in both directions. Each path connects two distinct attractions. We can move from any attraction to any other attractions by passing through one or more paths. Because JOI-kun and IOI-chan have the map of JOIOI park, they know how attractions are connected by paths.

Each attraction has a **message board**. On each message board, JOI-kun can write only one integer, either 0 or 1. He can write different integers on different message boards. Once he writes an integer on a message board, it will not be overwritten by other visitors.

JOI-kun and IOI-chan want to play with different attractions. Therefore, they will play separately for a while, and then, they will meet at some point. Since they do not have communication tools such as mobile phones, JOI-kun decided to use message boards to tell an integer X, which describes a time to meet, to IOI-chan.

In concrete terms, JOI-kun and IOI-chan will communicate by the following way:

- First, JOI-kun writes an integer, either 0 or 1, on every message board.
- Then, IOI-chan repeats moving from an attraction to another attraction by passing through a path between them. During this process, she will read the integer written on the message board of each attraction she visits (including the message board of her initial attraction).

By small number of moves, IOI-chan wants to know the integer X which JOI-kun wants to tell.

Task

Write two programs which enable for JOI-kun to tell the integer X to IOI-chan.

- Given the information on the paths in JOIOI park and the integer X, the first program decides an integer, either 0 or 1, which will be written by JOI-kun on each message board.
- Given the information on the paths in JOIOI park, the place of IOI-chan's initial attraction, and the integer written by JOI-kun on the message board on the initial attraction, the second program calculates the integer X by moving from an attraction to another attraction by passing through a path and by reading an integer on the message board of the destination.

Note that two programs will be given exactly the same information about the paths in JOIOI park. In particular, the numbers representing the attractions given to the two programs are the same. Also, the orders of the paths given to the two programs are the same.

Implementation Details

You need to submit two files.

The first file is `Joi.cpp`. This file implements JOI-kun's strategy, and implements the following function. The program should include `Joi.h`.

- `void Joi(int N, int M, int A[], int B[], long long X, int T)`

  For each test case, this function is called only once.

  - The parameter `N` is the number of attractions in JOIOI park.
  - The parameter `M` is the number of paths connecting attractions.
  - The parameters `A[]`, `B[]` are sequences of length M describing information of the paths connecting attractions. The elements `A[i]`, `B[i]` ($0 \le$ `i` $\le M - 1$) mean there is a path directly connecting the attraction `A[i]` and the attraction `B[i]`.
  - The parameter `X` is the integer which JOI-kun wants to tell IOI-chan.
  - The parameter `T` is the subtask number of the test case.

  Your program should call the following function:

  - ★ `void MessageBoard(int attr, int msg)`

    This function writes an integer to a message board.

    - ◇ The parameter `attr` is the number of attraction of the message board. `attr` is the number of attraction of the message board. `attr` should be an integer greater than or equal to 0 and less than or equal to $N - 1$. If it is not satisfied, your program is considered as **Wrong Answer[1]**. Your program cannot call this function with the same parameter `attr`. If this function is called twice with the same parameter `attr`, your program is considered as **Wrong Answer[2]**.
    - ◇ The parameter `msg` is an integer which will be written on the message board of the attraction `attr`. `msg` is an integer, either 0 or 1. If it is not satisfied, your program is considered as **Wrong Answer[3]**.

  Your program should call the function `MessageBoard` exactly N times. When function `Joi` terminates, if the number of times the function `MessageBoard` is called is different from N, your program is considered as **Wrong Answer[4]**.

  Your program is terminated if the function call by `Joi` is considered invalid.

The second file is `Ioi.cpp`. This file implements IOI-chan's strategy, and implements the following function. The program should include `Ioi.h`.

- `long long Ioi(int N, int M, int A[], int B[], int P, int V, int T)`

  For each test case, this function is called only once.

- ○ The parameter `N` is the number of attractions in JOIOI park.
- ○ The parameter `M` is the number of paths connecting attractions.
- ○ The parameters `A[]`, `B[]` are sequences of length M describing information of the paths connecting attractions. The elements `A[i]`, `B[i]` ($0 \leq \texttt{i} \leq M - 1$) mean there is a path directly connecting the attraction `A[i]` and the attraction `B[i]`.
- ○ The parameter `P` is the number of IOI-chan's initial attraction.
- ○ The parameter `V` is the integer written by JOI-kun on the message board of the attraction `P`.
- ○ The parameter `T` is the subtask number of the test case.

The parameters `N`, `M`, `A[]`, `B[]`, `T` given to the function `Ioi` are the same as the parameters given to the function `Joi`.

The function `Ioi` should return the integer which JOI-kun wants to tell (namely, the parameter `X` given to the function `Joi`). If it returns a different value, your program is considered as **Wrong Answer[5]**.

Your program can call the following function:

★ `int Move(int dest)`

This function describes IOI-chan's move.

- ◇ The parameter `dest` is the number of the attraction which is the destination of IOI-chan. `dest` should be an integer greater than or equal to 0 and less than or equal to $N - 1$. If it is not satisfied, your program is considered as **Wrong Answer[6]**. The attraction `dest` should be directly connected with IOI-chan's current attraction by a path. If it is not satisfied, your program is considered as **Wrong Answer[7]**.
- ◇ The return value of this function is the integer written on the message board of the attraction `dest` by JOI-kun.

Your program cannot call the function `Move` more than 20 000 times. If it is not satisfied, your program is considered as **Wrong Answer[8]**.

Important Notices

- Your program can implement other functions for internal use, or use global variables. Submitted programs will be compiled with the grader, and become a single executable file. All global variables and internal functions should be declared `static` to avoid confliction with other files. Since the programs of JOI-kun and IOI-chan will be executed as 2 distinct processes when they are graded, they can not share global variables.
- Your program should not use standard input and standard output. Your program should not communicate with other files by any methods.

Compilation and Test Run

You can download an archive file from the contest webpage which contains a sample grader to test your program. The archive file also contains a sample source file of your program.

A sample grader consists of one source file which is `grader.cpp`. For example, if your programs are `Joi.cpp` and `Ioi.cpp`, you run the following commands to compile your programs.

```
g++ -std=c++14 -O2 -o grader grader.cpp Joi.cpp Ioi.cpp
```

When the compilation succeeds, the executable file `grader` is generated.

Note that the actual grader is different from the sample grader. The sample grader will be executed as a single process, which will read input data from the standard input and write the results to the standard output.

Input for the sample grader

The sample grader reads the following data from the standard input.

- The first line of input contains five space separated integers N, M, X, P, T. This means there are N attractions, M paths, JOI-kun wants to tell the integer X to IOI-chan, IOI-chan's initial attraction is P, and the subtask number of this test case is T.
- The $(\mathtt{i}+1)$-th line $(0 \leq \mathtt{i} \leq M-1)$ of the following M lines contains two space separated integers `A[i]` and `B[i]`. This means there is a path directly connecting the attraction `A[i]` and the attraction `B[i]`.

Output of the sample grader

When the program terminates successfully, the sample grader writes the following information to the standard output. (The quotation mark is not written actually.)

- If your program is considered as correct, the sample grader reports the number of moves of IOI-chan in the following form "`Accepted : #move=12345`."
- If your program is considered as Wrong Answer, the sample grader writes its type in the following form "Wrong Answer [1]."

If your program is considered as several types of Wrong Answer, the sample grader reports only one of them.

Constraints

All input data satisfy the following conditions. See section of 'Input for the sample grader' for the meaning of the variables N, M, `A[i]`, `B[i]`, P, X.

- $60 \le N \le 10\,000$.
- $1 \le M \le 20\,000$.
- $0 \le \texttt{A[i]} \le N - 1$ $(0 \le \texttt{i} \le M - 1)$.
- $0 \le \texttt{B[i]} \le N - 1$ $(0 \le \texttt{i} \le M - 1)$.
- $\texttt{A[i]} \ne \texttt{B[i]}$ $(0 \le \texttt{i} \le M - 1)$.
- $(\texttt{A[i]}, \texttt{B[i]}) \ne (\texttt{A[j]}, \texttt{B[j]})$ $(0 \le \texttt{i} < \texttt{j} \le M - 1)$.
- $(\texttt{A[i]}, \texttt{B[i]}) \ne (\texttt{B[j]}, \texttt{A[j]})$ $(0 \le \texttt{i} < \texttt{j} \le M - 1)$.
- We can move from any attraction to any other attractions by passing through one or more paths.
- $0 \le X \le 2^{60} - 1$.
- $0 \le P \le N - 1$.

Subtask

There are 5 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [8 points]

- $T = 1$.
- $N \le 300$.

Subtask 2 [10 points]

- $T = 2$.

Subtask 3 [10 points]

- $T = 3$.
- $M = N - 1$.
- $\texttt{A[i]=i}, \texttt{B[i]=i+1}$ $(0 \le \texttt{i} \le N - 2)$.
- Your program cannot call the function `Move` more than 250 times.

Subtask 4 [55 points]

- $T = 4$.
- $240 \leq N$.

The score of this subtask is calculated as follows:

- Let C be the maximum number of calls to the function `Move` by your program for every test case in this subtask.
- The score of this subtask is
  - 0 point if $960 < C$.
  - $\left\lfloor 55 - 13 \log_2 \left(\frac{C}{120} \right) \right\rfloor$ points if $120 < C \leq 960$. (Here, $\lfloor x \rfloor$ means the largest integer not exceeding x.)
  - 55 points if $C \leqq 120$.

  Note that, when $960 < C$, in the contest system, the details of this subtask may be "Correct : 0 point" or "Incorrect."

Subtask 5 [17 points]

- $T = 5$.
- Your program cannot call the function `Move` more than 120 times.

Sample Communication

Here is a sample input for grader and corresponding function calls. Note that we only give a part of the sample input because it has large size. For a complete form of the sample input, see `sample-01.txt`, which is available on the contest website.

Sample Input	Sample Calls			
	Call	Return	Call	Return

`60 59 123 5 1`
`0 1`
`1 2`
`2 3`
`3 4`
`4 5`
`5 6`
`6 7`
`7 8`
`8 9`
`9 10`
`...`
	`Joi(...)`			

			`MessageBoard(0,0)`	
			`MessageBoard(1,1)`	
			`MessageBoard(2,1)`	
			`MessageBoard(3,0)`	
			`MessageBoard(4,0)`	
			`MessageBoard(5,1)`	
			`...`	
	`Ioi(...)`			
			`Move(4)`	0
			`Move(3)`	0
			`Move(2)`	1
			`Move(3)`	0
		123		

Here the parameters for `Joi(...)`, `Ioi(...)` are as follows.

Parameter	`Joi(...)`	`Ioi(...)`
`N`	60	60
`M`	59	59
`A`	$\{0, 1, 2, \ldots, 57, 58\}$	$\{0, 1, 2, \ldots, 57, 58\}$
`B`	$\{1, 2, 3, \ldots, 58, 59\}$	$\{1, 2, 3, \ldots, 58, 59\}$
`X`	123	
`P`		5
`V`		1
`T`	1	1