City

There are many cities in the Kingdom of JOI. The road network satisfies the following conditions:

- (Condition 1) The cities are numbered from 0 to $N - 1$. Here, N is the number of cities in the Kingdom of JOI.
- (Condition 2) The cities are connected by $N - 1$ roads. We can pass each road in both directions. We can travel from every city to any other cities if we pass through several roads.
- (Condition 3) We can travel from the city 0 to every city by passing at most 18 roads.

Every day, in the Kingdom of JOI, many people depart from the city 0 to other cities. Because many people have two destinations, they sometimes ask the following queries: for two different cities X, Y, which one of (0), (1), (2) is satisfied?

(0) We necessarily pass the city Y when we travel from the city 0 to the city X.

(1) We necessarily pass the city X when we travel from the city 0 to the city Y.

(2) Neither (0) nor (1).

Note that, in the above situation, exactly one of (0), (1), (2) is satisfied. When $X = 0$, we consider (1) is satisfied regardless of the value of Y. Similarly, when $Y = 0$, we consider (0) is satisfied regardless of the value of X.

It is known that, just like the road network in the Kingdom of JOI, the above Conditions 1–3 are satisfied in other countries as well. In the Kingdom of JOI, people plan to develop the following two machines so that they can be used also in other countries.

- (Machine 1) Given the number of cities N and information of the road network, this machine assigns a code to each city. A code is an integer between 0 and $2^{60} - 1$, inclusive.
- (Machine 2) Given the codes of two different cities X, Y assigned by Machine 1, this machine answers the query.

If large integers are assigned as codes, it is difficult to treat them. We want to develop machines so that smaller values are assigned as codes.

Note that, when Machine 2 is used, neither the number of cities N nor information of the road network is directly given to the machine.

Task

In order to develop two machines as above, write the following two programs:

- Given the number of cities N in the Kingdom of JOI and information of the road network, the first program assigns a code to each city.
- Given the codes of two different cities assigned by the first program, the second program answers the query for these cities.

Implementation Details

You need to submit two files written by the *same programming language*.

The first file is either `Encoder.c` or `Encoder.cpp`. This file implements Machine 1. It should implement the following function. The program should include `Encoder.h`.

- `void Encode(int N, int A[], int B[])`

  For each test case, this function is called once.

  - The parameter `N` is the number of cities N.
  - The parameters `A[]`, `B[]` are integer sequences of length $N-1$ describing the road network. The elements `A[i]`, `B[i]` ($0 \leq$ `i` $\leq N-2$) mean there is a road connecting the city `A[i]` and the city `B[i]` directly.

  Your program should call the following function.

  ★ `void Code(int city, long long code)`

  This function assigns codes to the cities.

  - ◇ The parameter `city` is the city to which a code is assigned. The parameter `city` should be an integer between 0 and $N-1$, inclusive. If the call to this function has parameters outside this range, your program is considered as **Wrong Answer[1]**. It should not be called twice with the same parameters. If it is called twice with the same parameters, your program is considered as **Wrong Answer[2]**.
  - ◇ The parameter `code` is the code assigned to the city whose number is `city`. The parameter `code` should be an integer between 0 and $2^{60}-1$, inclusive. If the call to this function has parameters outside this range, your program is considered as **Wrong Answer[3]**.

  Your program should call the function `Code` exactly N times. If the number of calls to `Code` is not N when the function `Encode` is terminated, your program is considered as **Wrong Answer[4]**.

  If the call to the function `Encode` is considered Wrong Answer, your program is terminated immediately.

The second file is either `Device.c` or `Device.cpp`. This file implements Machine 2. It should implement the following function. The program should include `Device.h`.

- `void InitDevice()`

  This function initializes Machine 2. The function `InitDevice` is called once before calls to the following function `Answer`.

- `int Answer(long long S, long long T)`

  This function answers each query. For each query, the function `Answer` is called once.

  - The parameters `S`, `T` are the codes assigned to two different cities X, Y by the function `Encode`.
  - To answer the query, the return value of the function `Answer` should satisfy the following conditions:
    - The return value should be `0` if we necessarily pass the city Y when we travel from the city 0 to the city X.
    - The return value should be `1` if we necessarily pass the city X when we travel from the city 0 to the city Y.
    - Otherwise, the return value should be `2`.

  Hence the return value of the function `Answer` should be an integer between 0 and 2, inclusive. If the return value is outside this range, your program is considered as **Wrong Answer[5]**. If the return value is inside this range but it does not satisfy the above conditions, your program is considered as **Wrong Answer[6]**.

Grading Procedure

The grading is done in the following way. If your program is considered as Wrong Answer, it is terminated immediately.

(1) The function `Encode` is called once.

(2) The function `InitDevice` is called once.

(3) For each test case, there are Q queries given to Machine 2. For the j-th query ($1 \leq j \leq Q$), the function `Answer` is called whose parameter `S` is S_j and parameter `T` is T_j, where S_j, T_j are the codes assigned to the cities X_j, Y_j by the function `Encode`, respectively.

(4) Your program is considered as correct.

Important Notices

- Running time and memory usage are calculated for (1), (2), (3) of Grading Procedure. Note that, in the procedure (3), the function `Answer` is called Q times in total.
- Your program must not be considered as Wrong Answer in the function call to `Encode` in the procedure (1), `InitDevice` in the procedure (2), or `Answer` in the procedure (3). Your program must be executed without runtime error.
- 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. The programs of Machine 1 and Machine 2 will be executed as two 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 either `grader.c` or `grader.cpp`. For example, if your programs are `Encoder.c` and `Device.c`, or `Encoder.cpp` and `Device.cpp`, you run the following commands to compile your programs.

- C

```
gcc -std=c11 -O2 -o grader grader.c Encoder.c Device.c -lm
```

- C++

```
g++ -std=c++14 -O2 -o grader grader.cpp Encoder.cpp Device.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 contains two space separated integers N, Q. This means there are N cities, and Q queries are given.

- The $(i+1)$-th line $(0 \leq i \leq N-2)$ of the following $N-1$ lines contains two space separated integers A_i, B_i. This means there is a road connecting the city A_i and the city B_i directly.
- The j-th line $(1 \leq j \leq Q)$ of the following Q lines contains three space separated integers X_j, Y_j, E_j. This means $X = X_j$ and $Y = Y_j$ in the j-th query, and your program will be considered Wrong Answer by the sample grader if the answer to this query is different from E_j.

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 writes the maximum of the codes assigned to the cities in the following form "`Accepted : max_code=123456`."
- 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. For the meaning of N, Q, A_i, B_i, X_j, Y_j, see Input for the Sample Grader.

- $2 \leq N \leq 250\,000$.
- $1 \leq Q \leq 250\,000$.
- $0 \leq A_i \leq N-1$ $(0 \leq i \leq N-2)$.
- $0 \leq B_i \leq N-1$ $(0 \leq i \leq N-2)$.
- $A_i \neq B_i$ $(0 \leq i \leq N-2)$.
- We can travel from the city 0 to every city by passing at most 18 roads.
- $0 \leq X_j \leq N-1$ $(1 \leq j \leq Q)$.
- $0 \leq Y_j \leq N-1$ $(1 \leq j \leq Q)$.
- $X_j \neq Y_j$ $(1 \leq j \leq Q)$.

Subtask

There are 2 subtasks. The score and additional constraints of each subtask are as follows:

Subtask 1 [8 points]

- $N \leq 10$.

Subtask 2 [92 points]

There are no additional constraints. In this subtask, the score is calculated as follows:

- Let L be the maximum of the codes assigned to the cities in all test cases for this subtask.
- The score for this subtask is:
  - 0 point if $2^{38} \leq L$.
  - 10 points if $2^{36} \leq L \leq 2^{38} - 1$.
  - 14 points if $2^{35} \leq L \leq 2^{36} - 1$.
  - 22 points if $2^{34} \leq L \leq 2^{35} - 1$.
  - $\lfloor 372 - 10 \log_2(L + 1) \rfloor$ points if $2^{28} \leq L \leq 2^{34} - 1$. Here, $\lfloor x \rfloor$ is the largest integer not exceeding x.
  - 92 points if $L \leq 2^{28} - 1$.

  Note that, if $2^{38} \leq L$, the result for this subtask may be displayed as "Accepted : 0 points" or "Wrong Answer" in the contest system,

Sample Communication

Here is a sample input for sample grader and corresponding function calls.

Sample Input	Sample Calls	
6 5 4 1 0 3 4 5 3 2 3 4 2 4 2 1 0 0 5 1 2 5 3 0 4 1 1	Machine 1	Machine 2
	`Encode(...)`	
	`Code(0,0)`	
	`Code(2,4)`	
	`Code(4,16)`	
	`Code(1,1)`	
	`Code(3,9)`	
	`Code(5,25)`	
		`InitDevice()`
		`Answer(4,16)`
		`Answer(1,0)`
		`Answer(25,1)`
		`Answer(25,9)`
		`Answer(16,1)`

The function `Encode(...)` is called with the following parameters:

Parameter	`Encode(...)`
`N`	6
`A`	$\{4, 0, 4, 3, 3\}$
`B`	$\{1, 3, 5, 2, 4\}$