Space Thief

You are active as a thief in JOI galaxy.

There are N stars numbered from 0 to $N - 1$ in JOI galaxy. There are M warp devices numbered from 0 to $M - 1$ in JOI galaxy. The warp device i ($0 \leq i \leq M - 1$) connects star U_i and star V_i bidirectionally. It is possible to travel from any star to any star by using warp devices.

A key is hidden in a certain star, and a treasure box is hidden in another certain star. Your mission is to specify numbers, the number of the star in which the key is hidden and in which the treasure box is hidden. To achieve your mission, you can ask questions up to 300 times as below.

- Orient each warp device. Specifically, for each warp device i ($0 \leq i \leq M - 1$), choose one of the following:
  - Allow travel only from star U_i to star V_i.
  - Allow travel only from star V_i to star U_i.
- Under these conditions, ask whether it is possible to travel from the star in which the key is hidden to the star in which the treasure box is hidden by using warp devices.

You want to specify numbers, the number of the star A in which the key is hidden and the star B in which the treasure box is hidden. To achieve a higher evaluation, you want to reduce the number of questions asked.

Given information about the galaxy, write a program that determines the star A in which the key is hidden and the star B in which the treasure box is hidden, by asking questions.

Implementation Details

You need to submit one file.

The file is `thief.cpp`. It should implement the following function. The program should include `thief.h` using the preprocessing directive `#include`.

- `void solve(int N, int M, std::vector<int> U, std::vector<int> V)`
  This function is called only once for each test case.
  - The parameter `N` is the number of stars N.
  - The parameter `M` is the number of warp devices M.
  - The parameters `U`, `V` are arrays of length M. For each i ($0 \leq i \leq M - 1$), `U[i]` and `V[i]` represent the stars U_i and V_i connected bi-directionally by the warp device i.

Within `thief.cpp`, your program can call the following function:

★ `int query(std::vector<int> x)`
Using this function, you can ask a question.

- ◇ The parameter `x` is an array of length M. For $0 \leq i \leq M - 1$:
  - ∗ If `x[i]` $= 0$, it means to allow travel only from star U_i to star V_i.
  - ∗ If `x[i]` $= 1$, it means to allow travel only from star V_i to star U_i.
- ◇ The return value is 0 or 1:
  - ∗ 0 means that it is not possible to travel from the star A to the star B by using warp devices.
  - ∗ 1 means that it is possible to travel from the star A to the star B by using warp devices.
- ◇ The length of the parameter `x` should be M. If not, your program is judged as **Wrong Answer [1]**.
- ◇ Each element of the parameter `x` should be 0 or 1. If not, your program is judged as **Wrong Answer [2]**.
- ◇ You must not call function `query` more than 300 times. If it is called more than 300 times, your program is judged as **Wrong Answer [3]**.

★ `void answer(int A, int B)`
You call this function to answer the number of the star A in which the key is hidden and the star B in which the treasure box is hidden.

- ◇ The parameter `A` represents that the key is hidden in the star A.
- ◇ The parameter `A` must be between 0 and $N - 1$, inclusive. If not, your program is judged as **Wrong Answer [4]**.
- ◇ The parameter `B` represents that the treasure box is hidden in the star A.
- ◇ The parameter `B` must be between 0 and $N - 1$, inclusive. If not, your program is judged as **Wrong Answer [5]**.
- ◇ If you answer a wrong number, your program is judged as **Wrong Answer [6]**.
- ◇ The function `answer` should be called exactly once. If the function `answer` is called more than once, your program is judged as **Wrong Answer [7]**. When the function `solve` terminates, if the function `answer` is not called, your program is judged as **Wrong Answer [8]**.

Important Notices

- Your program can implement other functions for internal use, or use global variables.
- Your program must not use the standard input and the standard output. Your program must not communicate with other files by any methods. However, your program may output debugging information to the standard error.

Compilation and Test Run

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

The sample grader is the file `grader.cpp`. In order to test your program, put `grader.cpp`, `thief.cpp`, `thief.h` in the same directory, and run the following command to compile your programs.

```
g++ -std=gnu++20 -O2 -o grader grader.cpp thief.cpp
```

Instead, you may run compile.sh contained in the archive file. In this case, run the following command to compile your programs.

```
./compile.sh
```

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.

$N\ M\ A\ B$
$U_0\ V_0$
$U_1\ V_1$
$\vdots$
$U_{M-1}\ V_{M-1}$

Output of the Sample Grader

The sample grader outputs the following information to the standard output (quotes for clarity).

- If your program is judged as correct, it reports the number of function calls to `query` as "`Accepted: 25`".
- If your program is judged as any type of Wrong Answer, the sample grader writes its type as "`Wrong Answer [4]`".

If your program satisfies the conditions of several types of Wrong Answer, the sample grader reports only one of them. The sample grader may terminate the execution when one of the conditions for wrong answer is met.

Notices for Grading

For some of the test cases, the actual grader is adaptive. This means the grader does not have a fixed answer in the beginning, and responds according to previous function calls to `query`. It is guaranteed that there is at least one answer which does not contradict all the responses of the the grader.

Constraints

All the input data satisfy the following conditions.

- $2 \leq N \leq 10\,000$.
- $1 \leq M \leq 15\,000$.
- $0 \leq A \leq N - 1$.
- $0 \leq B \leq N - 1$.
- $A \neq B$.
- $0 \leq U_i < V_i \leq N - 1$ $(0 \leq i \leq M - 1)$.
- $(U_i, V_i) \neq (U_j, V_j)$ $(0 \leq i < j \leq M - 1)$.
- It is possible to travel from any star to any star by using warp devices.

Subtasks

1. (7 points) $M = N - 1,\ \ U_i = i,\ \ V_i = i + 1\ (0 \leq i \leq M - 1)$.
2. (13 points) $M = N - 1,\ \ U_i = 0,\ \ V_i = i + 1\ (0 \leq i \leq M - 1)$.
3. (2 points) $M = N - 1,\ \ N \leq 8$.
4. (8 points) $M = N - 1,\ \ N \leq 50$.
5. (5 points) $M = N - 1,\ \ N \leq 150$.
6. (5 points) $M = N - 1,\ \ N \leq 250$.
7. (40 points) $M = N - 1$. In this subtask, points are awarded as follows:
   - If any test case in Subtask 7 is judged as wrong answer or exceeds the time limit, memory limit, or encounters a runtime error, the subtask score is 0 points.
   - If not, let T denote maximum value of the number of function calls to `query` in all test cases of this subtask. The subtask score is determined as follows:
     - 20 points if $120 < T$.
     - 30 points if $70 < T \leq 120$.
     - 40 points if $T \leq 70$.
8. (20 points) No additional constraints. In this subtask, points are awarded as follows:
   - If any test case in Subtask 8 is judged as wrong answer or exceeds the time limit, memory limit, or encounters a runtime error, the subtask score is 0 points.
   - If not, let T denote maximum value of the number of function calls to `query` in all test cases of this subtask. The subtask score is determined as follows:
     - 10 points if $120 < T$.
     - 15 points if $70 < T \leq 120$.
     - 20 points if $T \leq 70$.

The scores for Subtasks $1, 2, 3, 4, 5, 6$ do not depend on the number of function calls to `query`, as long as it does not exceed 300. However, when it is more than 70, the contest system might display “Output is partially correct”.

Sample Communication

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

Sample	Sample Function Calls		
Input 1	Function Calls	Function Calls	Return Values
`5 4 0 4`	`solve(5, 4, [0, 0, 1, 1], [1, 3, 2, 4])`		
`0 1`		`query([0, 1, 0, 0])`	`1`
`0 3`		`query([1, 1, 1, 0])`	`0`
`1 2`		`query([0, 0, 1, 0])`	`1`
`1 4`		`query([0, 0, 1, 1])`	`0`
		`answer(0, 4)`	

Choices corresponding to the 1-st function call to `query` are as below:

- Warp device 0 allows travel only from star 0 to star 1.
- Warp device 1 allows travel only from star 3 to star 0.
- Warp device 2 allows travel only from star 1 to star 2.
- Warp device 3 allows travel only from star 1 to star 4.

Under these conditions, it is possible to travel from star 0 to star 4 by using warp devices 0 and 3 in this order, so the return value is 1.

Choices corresponding to the 2-nd function call to `query` are as below:

- Warp device 0 allows travel only from star 1 to star 0.
- Warp device 1 allows travel only from star 3 to star 0.
- Warp device 2 allows travel only from star 2 to star 1.
- Warp device 3 allows travel only from star 1 to star 4.

Under these conditions, it is not possible to travel from star 0 to star 4 by using warp devices, so the return value is 0.

Choices corresponding to the 3-rd function call to `query` are as below:

- Warp device 0 allows travel only from star 0 to star 1.
- Warp device 1 allows travel only from star 0 to star 3.
- Warp device 2 allows travel only from star 2 to star 1.
- Warp device 3 allows travel only from star 1 to star 4.

Under these conditions, it is possible to travel from star 0 to star 4 by using warp devices, so the return value is 1.

Under the conditions of the choices corresponding to the 4-th function call to `query`, it is not possible to travel from 0 to star 4 by using warp devices, so the return value is 0.

The function call to `answer` represents that you answer that star A is star 0, and star B is star 4.

This Sample Input satisfies the constraints of Subtasks $3, 4, 5, 6, 7, 8$. Among the files which can be obtained from the contest webpage, `sample-01-in.txt` corresponds to Sample Input 1.

Also, the function calls in a sample source file of your program contained in the archive file from the contest webpage is the same as these function calls.