

# Problem Solving Practice

## Coding Problem #1

CSE4152  
Sogang University



# Finding Celebrities

A celebrity is defined as a person who is known by everybody but does not know anyone. How can you find any celebrities among  $N$  people?

You can freely ask anyone as follows: "Hey, you know Mr. (or Ms.) X?"

Suppose that there are  $N$  people. Design a method that

- a) determines whether there is some celebrities among the people
- b) if there is any celebrities, find them efficiently (minimizing the number of questions you make).

# Finding Celebrities

In this problem, you will only be working within the `main.cpp` file. The following two functions are provided for you to interact with the problem:

- `ask_a_to_know_b(int a, int b) :`
  - Ask person A if they know person B.
  - return 1 if A knows B, otherwise return 0.
  - Never call `ask_a_to_know_b(i, i)` (i.e., do not ask if someone knows themselves)
- `answer(int x) :`
  - verifies if person x is a celebrity
  - if there is no celebrity, call -1

Your task is to implement the logic within the `main()` function in `main.cpp`

# Finding Celebrities

## Interaction

### Your Output:

- You will print interactions using the provided `ask_a_to_know_b(a, b)` function to ask if one person knows another.
- Example: `? 1 2` asks if person 1 knows person 2.

### Judge's Response:

- The judge will respond with `1` (Yes) or `0` (No) based on the relationship.
- Example: `0` means person 1 does not know person 2 if the question is `? 1 2`.

### Final Answer:

- After determining the celebrity (or finding there is none), call `answer(x)` with the candidate's index or `-1`.
- Example: `! 1` means you believe person 1 is the celebrity. The judge will then confirm if this is correct.

# Finding Celebrities

## Steps to test your algorithm.

You should test your program by interacting with it by yourself.

### Step 1. Create your own celebrity graph

- Before diving into the code, start by visualizing the relationships as a graph.

### Step 2. Run the program and answer function `ask_a_to_know_b` by your hand.

- based on your graph, manually type the output of `ask_a_to_know_b(a, b)` for various pairs (a, b).
- For example, if Person 1 knows Person 2 (`ask_a_to_know_b(1, 2)`), draw an edge from 1 to 2.

### Step 3. Type the response for `answer(x)`

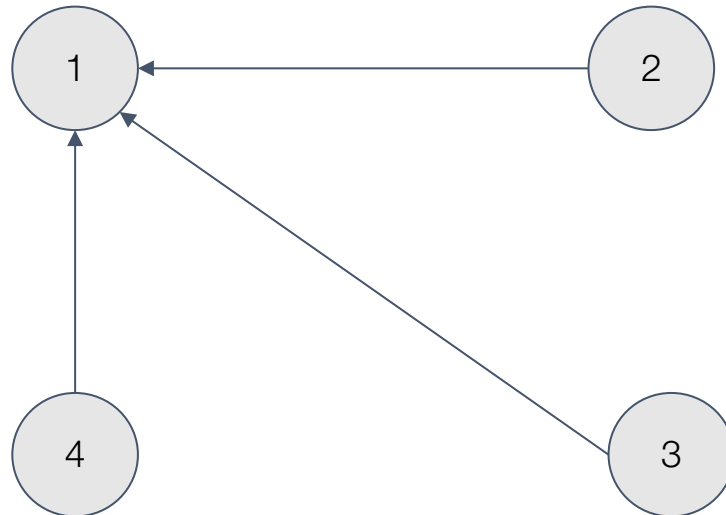
- Check if the celebrity identified in the code (`answer(x)`) matches the one you set in your created celebrity graph

# Finding Celebrities

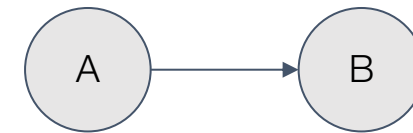
## Example

### Step 1. Create your own celebrity graph

- Before diving into the code, start by visualizing the relationships as a graph.



celebrity graph

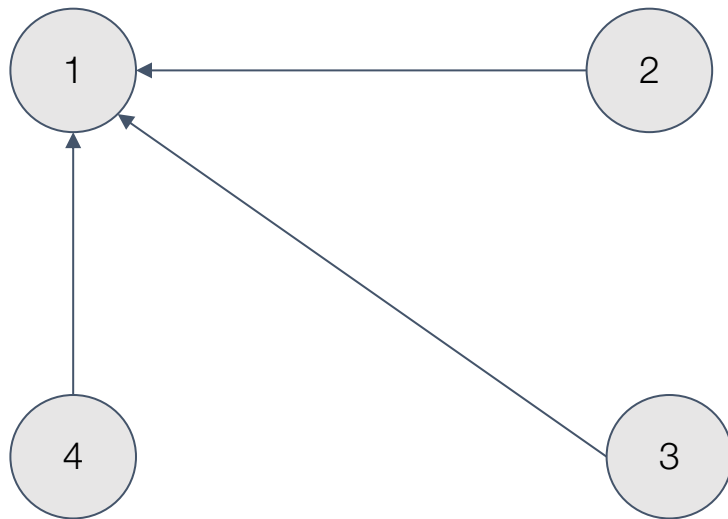


A knows B

# Finding Celebrities

## Example

Step 2. Reply to every call `ask_a_to_know_b` called by your program.



celebrity graph

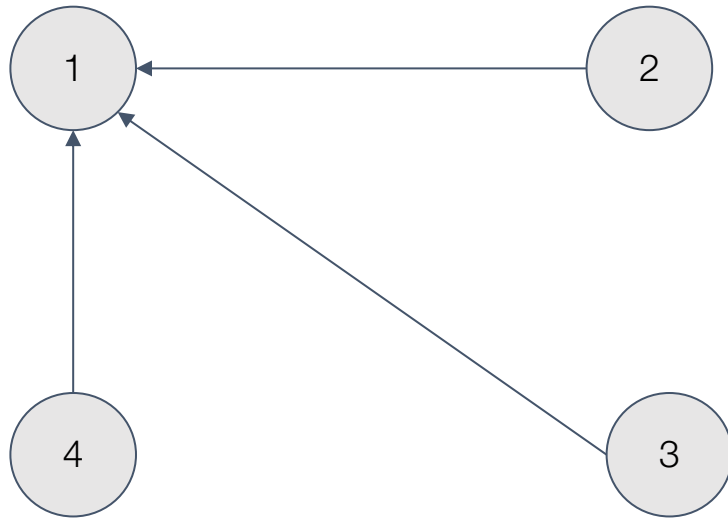
Your Output	Judge	Explanation
	4	N = 4.
? 1 2		Does 1 know 2?
	0	1 does not know 2.
? 2 1		Does 2 know 1?
	1	2 does know 1.
? 1 3		Does 1 know 3?
	0	1 does not know 3.
? 3 1		Does 3 know 1?
	1	3 does know 1.
? 1 4		Does 1 know 4?
	0	1 does not know 4.
? 4 1		Does 4 know 1?
	1	4 does know 1.
! 1		Is 1 the celebrity?
	1	1 is the celebrity.

Interaction

# Finding Celebrities

## Example

Step 3. Reply to `answer(x)` function



celebrity graph

Your Output	Judge	Explanation
	4	N = 4.
? 1 2		Does 1 know 2?
	0	1 does not know 2.
? 2 1		Does 2 know 1?
	1	2 does know 1.
? 1 3		Does 1 know 3?
	0	1 does not know 3.
? 3 1		Does 3 know 1?
	1	3 does know 1.
? 1 4		Does 1 know 4?
	0	1 does not know 4.
? 4 1		Does 4 know 1?
	1	4 does know 1.
! 1		Is 1 the celebrity?
	1	1 is the celebrity.

Interaction

1  
call **answer(1)**

answer

# Finding Celebrities

## Scoring

- The fewer questions you ask to identify the celebrity, the higher your score will be. Therefore, it's a good idea to design your code with an algorithm that finds the answer using as few questions as possible.
- The exact scoring formula is available on ELICE, so please refer to it for details.

# Minimum calculation

Imagine an electric calculator which has only two arithmetic operations, i.e., assignment operation ( $:=$ ) and multiplication operation ( $\times$ ). We want to let this machine compute  $b = a^n$  for given numbers  $a$  and  $n$ . Design a method that produce a code for this machine, minimizing the number of multiplications.

# Minimum calculation

Input

16

Output

4 1 2 4 8 16

# Minimum calculation

## Scoring

- The closer your solution is to the minimum number of steps, the higher your score will be. To achieve a higher score, please consider developing an algorithm that can produce the solution with the minimum number of steps
- The exact scoring formula is available on ELICE, so please refer to it for details.

# Submission

- Please submit your solution to this coding problem on the Elice platform by **Monday, September 23rd**.
- Late submissions will not be accepted, so make sure to submit on time
- Feel free to post any questions in the Q&A section on the Cyber Campus