MMCC '15 P1 - Inaho

Time limit: 1.4s **Memory limit:** 512M

Inaho Kaizuka has a bionic eye. It has a lot of software on it, including some "games". One game Inaho is particularly fond of is "Graph Simulator 2015". In this game, Inaho controls a graph with N vertices and initially no edges. He may choose to arbitrarily add a bidirectional edge between two vertices, remove the most recent edge added in the graph, or query the size of the connected component of a vertex. Unfortunately, the game Inaho currently has is broken — you need to write an exact copy of it.

Implementation Details

You should write a program which implements the bionic eye's software "Graph Simulator 2015". Locally, your program should include the header file `inaho.h` by `#include "inaho.h"`. When grading your submission online, the judge will automatically prepend this line to your file.

Your program should implement the following procedures.

- `void Init(int N)` This procedure is called only once in the beginning. The parameter N is the number of vertices Inaho is simulating. Use it to perform any initialization your program might need.
- `void AddEdge(int U, int V)` This procedure is called when Inaho adds an edge between two vertices U and V in his graph. It will hold that $1 \leq U, V \leq N, U \neq V$.
- `void RemoveLastEdge()` This procedure is called when Inaho removes the most recent edge added to the graph by `AddEdge` that has not already been removed. It is guaranteed that there is at least one edge in the graph when this procedure is called.
- `int GetSize(int U)` This procedure is called when Inaho queries for the size of the connected component of a vertex. You should return that size as an `int`. It will hold that $1 \leq U \leq N$.

You can find a sample grader in the C language here.

You can find the header `inaho.h` here.

You can find sample input for the sample grader with Windows-style line endings here.

Constraints

All input data satisfy the following conditions.

- $1 \leq N \leq 500\,000$.
- The total number of calls to `AddEdge`, `RemoveLastEdge`, and `GetSize` is less than or equal to $1\,000\,000$.

Note

The grader will generate correct input and output dynamically. This means that the time and memory on the submission results page may not necessarily be solely from your program's execution.