Classifica senza fili (classifica)

The web page for real-time ranking of the Italian Olympiads in Informatics has crashed, and the teachers are panicking as they are no more updated on how their favorite contestants are performing! Fortunately they know the ranking of the N contestants soon before the crash, so that the $(i+1)$-ranked contestant was `ids[`i`]`. From that moment on, they can only rely on leaks from the staff, profoundly moved by the situation. More precisely, there are leaks as soon as:

- contestant `id` overtakes the contestant immediately preceding her or him;
- contestant `id` is disqualified for having tried to hack the competition's system.[1]

The teachers, however, are confused by the large amount of data, and just want to know which contestant `id` appears in a queried position. Help them keeping track of all the leaks, so as to answer their queries.

Implementation

You shall submit one file having extension `.c`, `.cpp` or `.pas`.

> ☞ Among the attachments of this task you will find a template (`classifica.c`, `classifica.cpp`, `classifica.pas`) with a sample incomplete implementation.

You need to implement the following functions:

C/C++	`void inizia(int N, int ids[]);`
Pascal	`procedure inizia(N: longint; ids: array of longint);`

- N is an integer representing the number of contestants.
- The array `ids` is indexed from 0 to $N-1$ and contains the contestants ids (integers from 0 to $N-1$) in their position before the crash.

C/C++	`void supera(int id);`
Pascal	`procedure supera(id: longint);`

- `id` is an integer representing the id of the contestant that is currently overtaking.

C/C++	`void squalifica(int id);`
Pascal	`procedure squalifica(id: longint);`

- `id` is an integer representing the id of the disqualified contestant.

[1] The staff *always* find out quickly about every such attempt!

C/C++	`int partecipante(int pos);`
Pascal	`function partecipante(pos: longint): longint;`

- `pos` is an integer representing the position that a teacher would like to query.
- The function must return the `id` of the contestant in that position.

The grader will first call the function `inizia`, then the functions `supera`, `squalifica`, `partecipante` for an arbitrary number of times (and in any order) and will print the returned values to the output file (in the corresponding order).

Grader

In the directory for this problem there is a simplified version of the grader used during evaluation, which you can use to test your solutions locally. The sample grader reads data from `stdin`, calls the function that you should implement and writes to `stdout` in the following format.

The input file is made of $Q + 2$ lines, containing:

- Line 1: integers N and Q.
- Line 2: values `ids[`i`]` for $i = 0 \dots N - 1$.
- Lines $3, \dots, Q + 2$: the description of a leak or a query, so it can be:
  - `'s' id`: if `id` overtakes;
  - `'x' id`: if `id` is disqualified;
  - `'p' pos`: if the contestant in position `pos` is queried.

The output file is made of a single line, containing:

- Line 1: the values returned from calls to function `partecipante` separated by a space.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $1 \leq Q \leq 1\,000\,000$.
- $0 \leq$ `ids[`i`]` $\leq N - 1$ for all $i = 0 \dots N - 1$.
- `ids[`i`]` $\neq$ `ids[`j`]` for all $i \neq j$ (the numbers contained in `ids` are all distinct).
- $0 \leq$ `id` $\leq N - 1$ in calls to functions `supera` and `squalifica`.
- The contestant which is ranked first does not overtake.
- Once somebody is disqualified, he or she cannot overtake nor be further disqualified anymore.
- Position `pos` in calls to function `partecipante` always exists.

Scoring

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

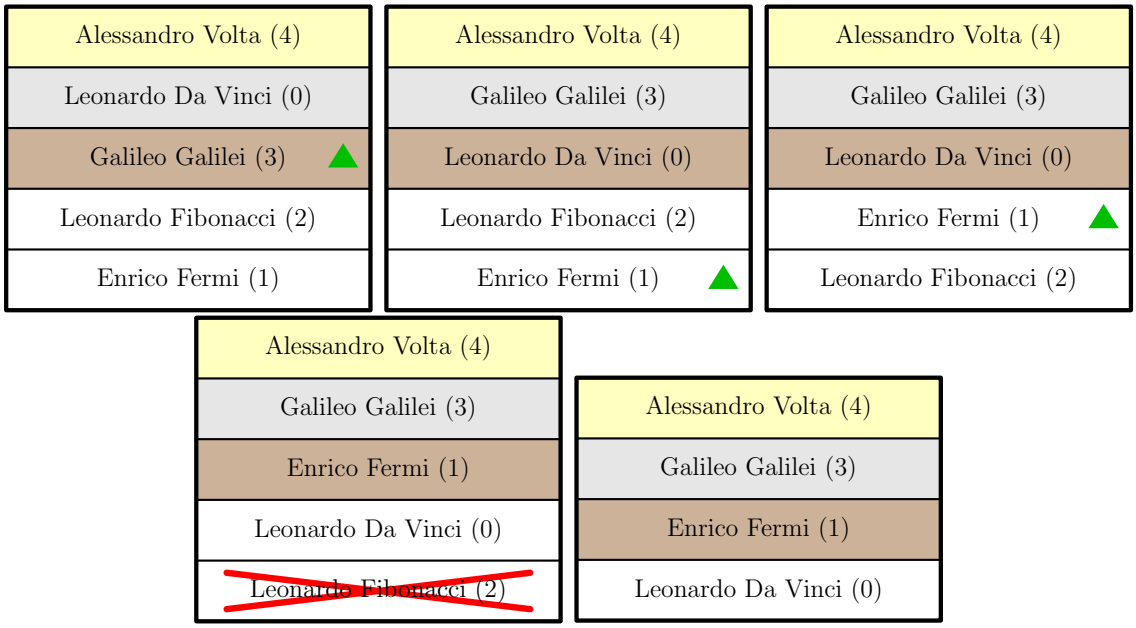
- **Subtask 1** [**0 punti**]: Examples.
- **Subtask 2** [**18 punti**]: $N, Q \leq 10\,000$.
- **Subtask 3** [**16 punti**]: There are no calls to function `squalifica`. Moreover $Q \leq 100\,000$.
- **Subtask 4** [**19 punti**]: The function `partecipante` will be called after all the calls to `supera` and `squalifica`. Moreover $Q \leq 100\,000$.
- **Subtask 5** [**17 punti**]: There are no calls to function `supera`.
- **Subtask 6** [**18 punti**]: $Q \leq 100\,000$.
- **Subtask 7** [**12 punti**]: No additional limitations.

Examples

5 6 4 0 3 2 1 s 3 s 1 s 1 p 3 x 2 p 4	1 0
7 11 5 2 6 3 4 1 0 x 5 p 1 x 2 p 1 p 2 s 3 p 1 p 4 x 0 s 1 p 3	2 6 3 3 1 1

Explanation

In the **first example**, the ranking evolves as shown below:



In the **second example**, the ranking evolves as shown below: