Circus Show (cannons)

The final of the Olympiads in Teams is finally here and William wants to celebrate this special occasion organizing a circus show! He does not have any experience in the sector though, thus he decided to contact experts from the *Organizing Improvised Shows* circus company. When Giorgio heard about the plan, he immediately started doubting the qualities of this group as they seem a bit... unorganized.

The main exhibition of the proposed show, which is also the most dangerous one, involves a *cannon man* who is repeatedly thrown by a series of cannons until he reaches the final destination. The group plans to have N cannons arranged in a line and start with the *cannon man* ready to be thrown by cannon 0; the final goal for him is to reach the last cannon $N-1$.

Every cannon points to another cannon (potentially to itself) such that when the *cannon man* reaches the cannon i he is immediately thrown to the cannon T_i.

Figure 1: One of the cannons used for the show.

While this sounds like a lot of fun, Giorgio was right: folks at *Organizing Improvised Shows* are very unorganized. After placing the cannons, they lost part of the original plan for setting the directions and now is unclear the trajectory the *cannon man* will take.

Giorgio wants to take charge of the situation to do a last-minute fix. After studying the current setting for every cannon, he is devising a plan to change some of them in order to make sure the *cannon man* is able to reach the final destination. Changing the setting for a cannon consists in modifying the value of T_i for that cannon and is known that modifying it to another value $T_i{}'$ will cost $|T_i{}' - i|$ units of time.

The show is about to start and Giorgio is running out of time: help him find the *minimum* total time required to change the setting of some of the cannons to let the *cannon man* reach cannon $N-1$ from cannon 0.

> ☞ Among the attachments of this task you may find a template file `cannons.*` with a sample incomplete implementation.

Input

The first line contains the only integer N, the number of cannons. The second line contains N integers T_i: the i-th one contains the index of the cannon to which the *cannon man* will be thrown if he lands on cannon i.

Output

You need to write a single line with an integer: the minimum total time required for making the changes so that the *cannon man* is able to reach cannon $N-1$.

Constraints

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq T_i < N$ for each $i = 0 \ldots N-1$.

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 points) Examples.

– **Subtask 2** (10 points) $N \leq 10$.

– **Subtask 3** (10 points) It is guaranteed that exists a solution the cost of which is *at most* one unit of time.

– **Subtask 4** (30 points) $N \leq 1000$.

– **Subtask 5** (20 points) $N \leq 6000$.

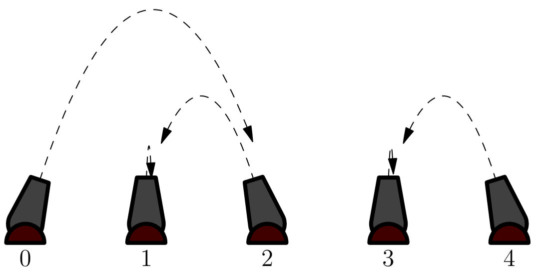
– **Subtask 6** (30 points) No additional limitations.

Examples

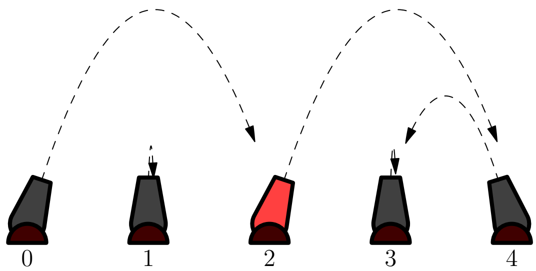
input	output
5 2 1 1 3 3	2
4 1 2 3 2	0

Explanation

In the **first sample case** a possible solution is to modify the setting of the third cannon, making it pointing directly towards the final destination. This only change has a cost of $4 - 2 = 2$ units of time.

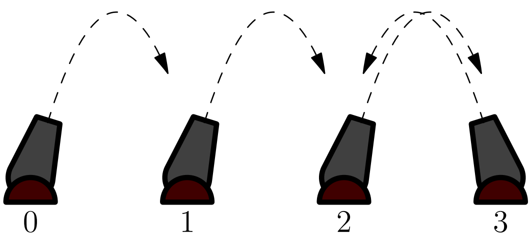


Initial situation.



After changing the third cannon.

In the **second sample case** the *cannon man* is already able to reach the final cannon without any modification to the current setting.



Initial situation.