

Removal

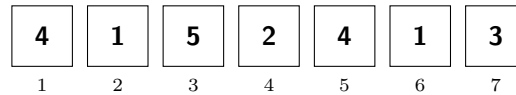
XIV OIJ, 2nd stage
June 10th, 2020

Code: **usu**
Time limit: **7 s**
Memory limit: **256 MB**



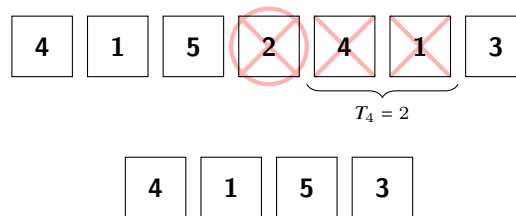
Bytie recently got an interesting game called *Removal*. It is a single player game in which the player uses a set of N blocks lined in a row from left to right. On each block there is some non-negative integer, on the i -th block on the left side that integer is T_i .

Consider an example set of $N = 7$ of blocks, with the following numbers written on them (4, 1, 5, 2, 4, 1, 3).

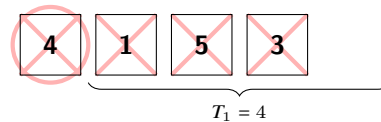


Each turn, Bytie chooses a certain block, removes it from the set, and together with it, removes as many neighbors on the right as the number on the selected block. So if she selects a block in the i -th position, she will also remove the T_i of the blocks on the right. If there are less than T_i blocks to the right of the selected element, then all the blocks on the right are removed. If, however, there are still some blocks left on the right side after removing the blocks, they are moved to the left so that a hole is not created in the sequence of blocks.

For example, if Bytie would choose a block in the fourth position with the number $T_4 = 2$, this block will be removed and the 2 blocks from its right side, i.e. the blocks in the fifth and sixth positions. Then Bytie will have the following blocks left: (4, 1, 5, 3) (as in the picture below).



Let us assume that now Bytie has chosen a block in the first position with the number $T_1 = 4$ on it. There are only 3 blocks on the right side, so Bytie can remove all the blocks.



The goal of the game is to remove all the blocks. Bytie has always been a supporter of the principle not to do only the amount of work that is necessary – she would like to achieve the goal of the game by making as few moves as possible. Will you help her in this task?

Write a program that reads the initial sequence of blocks in the game *Removal* and computes the minimum number of moves necessary to remove all the blocks.

Input

The first line of input contains one integer N ($1 \leq N \leq 200\,000$), specifying the number of blocks. The second (last) line of input contains N non-negative integers T_i ($0 \leq T_i < N$), separated by single spaces – the numbers written on the blocks from left to right.

Output

Output a single integer – the minimum number of moves needed to remove all the blocks.

Grading

The following table describes additional conditions that some test groups fulfill and the number of points that you can obtain by passing only the tests fulfilling these conditions.

Additional conditions	Points
only 0 and 1 appear on the blocks	29
$N \leq 10$	44
$N \leq 100$	55
$N \leq 1000$	67

Examples

Example input usu0a:

```
7
4 1 5 2 4 1 3
```

Example output usu0a:

```
2
```

Explanation: Byteie can start by selecting the block in the fourth position with 2 on it, and then select the block in the first position with 4 on it (according to the example described in the body above). Note that this is not the only possible optimal sequence of moves. You can also start by selecting the 5 block in the third position and then selecting the 4 block in the first position. Yet another option is to select the 4 block in fifth position and then the 4 block in first position.

Example input usu0b:

```
5
1 1 1 1 1
```

Example output usu0b:

```
3
```

Explanation: In this case, Byteie could always choose the first block, which would remove it, and the adjacent block to the right (if it exists). So, after the first move, the sequence of blocks is (1,1,1), after the second one: (1), and after the third one we don't have any blocks anymore.

Example input usu0c:

```
5
4 4 4 4 4
```

Example output usu0c:

```
1
```

Explanation: In this case, it is enough for Byteie to choose the first block. This will remove all the blocks.

Example input usu0d:

```
6
0 0 0 0 0 0
```

Example output usu0d:

```
6
```

Explanation: In this case, Byteie's choices do not matter, and she has to choose each block of this sequence getting (0,0,0,0,0), (0,0,0,0), (0,0,0), (0,0), (0) and an empty set of blocks.

Additional example tests

- test usu0e: $N = 990$, the sequence of numbers written on the blocks is (1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, ..., 44, 44, 44, 44)
- test usu0f: $N = 200\,000$, the sequence of numbers written on the blocks is (1, 0, 1, 0, 1, 0, ...)
- test usu0g: $N = 200\,000$, the sequence of numbers written on the blocks is (4, 1, 4, 1, 4, 1, ...)