

B. Bubble Sort Graph

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

Iahub recently has learned Bubble Sort, an algorithm that is used to sort a permutation with n elements a_1, a_2, \dots, a_n in ascending order. He is bored of this so simple algorithm, so he invents his own graph. The graph (let's call it G) initially has n vertices and 0 edges. During Bubble Sort execution, edges appear as described in the following algorithm (pseudocode).

```
procedure bubbleSortGraph()
    build a graph  $G$  with  $n$  vertices and 0 edges
    repeat
        swapped = false
        for  $i = 1$  to  $n - 1$  inclusive do:
            if  $a[i] > a[i + 1]$  then
                add an undirected edge in  $G$  between  $a[i]$  and  $a[i + 1]$ 
                swap(  $a[i]$ ,  $a[i + 1]$  )
                swapped = true
            end if
        end for
    until not swapped
    /* repeat the algorithm as long as swapped value is true. */
end procedure
```

For a graph, an independent set is a set of vertices in a graph, no two of which are adjacent (so there are no edges between vertices of an independent set). A maximum independent set is an independent set which has maximum cardinality. Given the permutation, find the size of the maximum independent set of graph G , if we use such permutation as the permutation a in procedure bubbleSortGraph.

Input

The first line of the input contains an integer n ($2 \leq n \leq 10^5$). The next line contains n distinct integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$).

Output

Output a single integer — the answer to the problem.

Examples

input
3 3 1 2
output
2

Note

Consider the first example. Bubble sort swaps elements 3 and 1. We add edge (1, 3). Permutation is now [1, 3, 2]. Then bubble sort swaps elements 3 and 2. We add edge (2, 3). Permutation is now sorted. We have a graph with 3 vertices and 2 edges (1, 3) and (2, 3). Its maximal independent set is [1, 2].