Problem B. Longest Increasing Subsequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	1024 mebibytes

Given an increasing integer sequence $A = a_1, a_2, \cdots, a_n$ of length n whose elements are all distinct, we generate another sequence B with the following algorithm.

Algorithm 1 Sequence Generating Algorithm

```
1:  function GENERATE(A)
2:      B ← A
3:      while true do
4:          B' ← B
5:          Let S be the sequence by sorting B in increasing order
6:          for i in [1, length of S) do
7:              if s_i + 1 ≠ s_{i+1} then                ▷ s_i is the i-th element in S
8:                  Add ⌊(s_i + s_{i+1})/2⌋ to the end of B'   ▷ ⌊x⌋ is the largest integer not larger than x
9:          if B = B' then
10:             break
11:         B ← B'
12:     return B
```

It is easy to prove that this algorithm will terminate and that elements of B are all distinct. Calculate the length of the longest increasing subsequence of B.

Input

There is only one test case in each test file.

The first line contains an integer n ($1 \le n \le 10^5$) indicating the length of sequence A.

The second line contains n integers $a_1, a_2, \cdots, a_n$ ($1 \le a_1 < a_2 < \cdots < a_n \le 10^{18}$) indicating the given sequence.

Output

Output one line containing one integer indicating the length of the longest increasing subsequence of B.

Example

standard input	standard output
3 1 5 20	11

Note

For the sample test case, $B = \{1, 5, 20, 3, 12, 2, 4, 8, 16, 6, 10, 14, 18, 7, 9, 11, 13, 15, 17, 19\}$. Its longest increasing subsequence is $\{1, 3, 4, 6, 7, 9, 11, 13, 15, 17, 19\}$ of length 11.