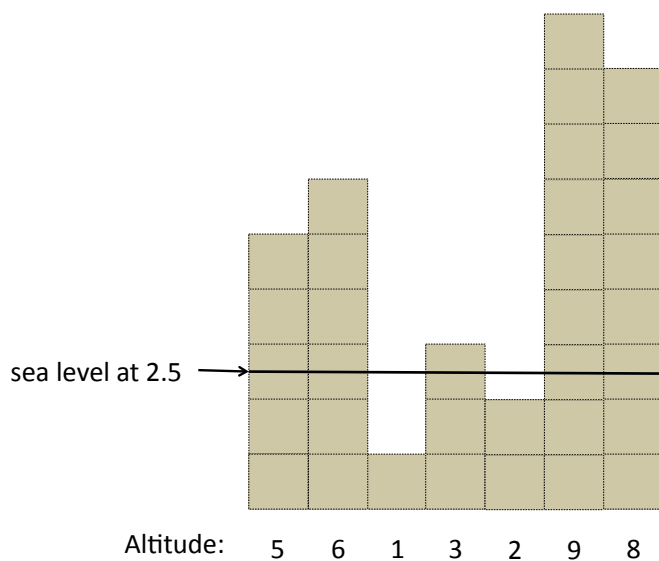


## Task 1: GLOBAL WARMING

A scientist wants to study how the rising sea level changes the landscape, in particular, how it changes the number of islands. He first investigates one-dimensional worlds. An one-dimensional world is represented by a sequence of non-negative integers  $\langle h_0, h_1, \dots, h_{n-1} \rangle$ , where each integer  $h_i$  is the altitude at the location  $i$ . The following figure depicts an example of such world represented by the sequence  $\langle 5, 6, 1, 3, 2, 9, 8 \rangle$ .



Now, if the sea level is at altitude 2.5, there are 3 islands formed by landmass of the first two columns, the fourth column and the last two columns. Furthermore, if the sea level is at altitude 3.5, there are only 2 islands. When the sea level is at altitude  $x$ , landmass with altitude  $x$  is considered to be submerged under the sea. Hence, if the sea level is at altitude 3, there are 2 islands. Note that having 3 islands is the maximum among all possible sea levels.

Given a one-dimensional world, the scientist wants to find the maximum number of islands among all sea levels.

### Input format

Your program must read from the standard input. The first line in the input contains the integer  $n$ , the total number of integers in the sequence. Next, it is followed by  $n$  lines where each line contains an integer. These  $n$  lines represent the sequence  $\langle h_0, h_1, \dots, h_{n-1} \rangle$ . All numbers in the sequence are non-negative and smaller than  $2^{30}$ . For the above example, the input is

7  
5  
6  
1  
3  
2  
9  
8

## Output format

Your program must write to the standard output an integer, which is the maximum number of islands. For the above example, the output is:

3

## Template

You may use the templates provided. The templates handle the input and output, but without the body of the following subroutines.

- **C program**

```
int gw (int N, int *H);
```

- **Pascal program**

```
function gw (N: LongInt; var H: array of LongInt): LongInt;
```

Each subroutine takes in two parameters  $N$  and  $H$ , where  $N$  is the size of the array, and  $H$  is the array representing the one-dimensional world.

## Subtasks

The maximum execution time on each input instance is 1.0 second. Your program will be tested on sets of input instances as follow:

1. (6 marks) All instances in this set satisfy  $N \leq 1,000$ .
2. (6 marks) All instances in this set satisfy  $N \leq 100,000$ . In addition, the altitude at each location is at most 20.
3. (7 marks) All instances in this set satisfy  $N \leq 100,000$ . In addition, the altitude at each location is unique, that is, no two numbers in the input sequence are the same.
4. (10 marks) All instances in this set satisfy  $N \leq 1,000,000$ . In addition, the numbers in the input sequence are unique.
5. (11 marks) All instances in this set satisfy  $N \leq 1,000,000$ .