Binary Search

Given a sorted array of n integers in which all elements appear twice (one after one) and one element appears only once. Implement an efficient algorithm to find that element. For example, if the given array is $[1, 1, 3, 3, 4, 5, 5, 7, 7, 8, 8]$, the output is 4, if the input array is $[1, 1, 3, 3, 4, 4, 5, 5, 7, 7, 8]$, the output is 8.

NOTE!!!: Your algorithm **MUST** have running time in $O(\log n)$, so simple scanning from left to right is not going to work.

Submit single file with implementation of your algorithm (`BinS.py` or `BinS.java`). Your implementation must read the given array from standard input and print the value of unique element in the array.

The input is provided on single line, such as for example

```
1 1 3 3 4 5 5 7 7 8 8
```

or

```
1 1 3 3 4 4 5 5 7 7 8
```

The output is

```
4
```

in the first case and

```
8
```

in the second case.

Test your solution carefully for all possible border cases, such as $n = 1$, unique element in the left-most or right-most position and so on ...

There is no advance knowledge of the size of the array or the range of values!!!