Problem H. Halve & Merge

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You have an array $a = (a_1, \ldots, a_n)$ that initially contains a permutation of numbers 1 through n. You have to process queries of two types:

- "1 p" ($1 \le p \le n$): find a_p in the current array a;
- "2 p" ($1 \le p \le n - 1$): replace a by the result of the function *merge* applied to arrays $(a_1, \ldots, a_p)$ and $(a_{p+1}, \ldots, a_n)$.

Function *merge* can be written in the following way.

```
func merge(var a as array, var b as array)
     var c as array
     while (a and b have elements)
          if (a[0] > b[0])
               add b[0] to the end of c
               remove b[0] from b
          else
               add a[0] to the end of c
               remove a[0] from a
     while (a has elements)
          add a[0] to the end of c
          remove a[0] from a
     while (b has elements)
          add b[0] to the end of c
          remove b[0] from b
     return c
```

Input

The first line contains two integers n and m — the length of the array and the number of queries ($2 \le n, m \le 2 \cdot 10^5$).

The second line contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$).

Each of the next m lines contains two integers t_i and p_i — the description of the i-th query ($t_i \in \{1, 2\}$, p satisfies the constraints given in the format description above).

Output

For each query of type 1, print the answer on a separate line.

Examples

standard input	standard output
4 3 4 3 2 1 2 1 2 1 1 2	1
5 7 4 3 5 2 1 2 4 2 1 1 3 1 1 2 4 1 4 1 5	3 1 3 5