Array Max

A is an array of *M* integers, initially all zero: *A[0], A[1], ..., A[M-1]*. The input describes a sequence of *N* assignments operations, each of the form "*A[i]=v*". After each assignment, you report the index *j* such that *A[j]* is the current maximum array value. In case of a tie, report the smallest such *j*.

You should produce the k^{th} output before reading the $(k+1)^{th}$ assignment.

Hints: we prefer solutions using O(M) space, where M is the array size. Note that M may be much smaller than N, the number of assignments. Refer to the Canvas Assignment to figure out how to do this, as well as suggestions on I/O to handle slow runtime.

Input Format

The first line has two integers:

M N

The next *N* lines each have two integers, describing an assignment "*A[i]=v*":

i v

Constraints

$1 \le M \le 50000$

$M \le N \le 500000$

$0 \le i < M$

$0 \le v < 10^9$

Output Format

There are *N* lines of output, corresponding to the *N* assignments. The k^{th} output is an integer *j*, such that *A[j]* is the largest array value after the k^{th} assignment. In case of a tie, report the smallest such *j*.

Sample Input 0

```
3 5
1 5
2 7
1 4
2 3
1 2
```

Sample Output 0

```
1
2
2
1
2
```

Explanation 0

After A[1]=5, the array is [0,5,0], and the max 5 is at 1.

After A[2]=7, the array is [0,5,7], and the max 7 is at 2.

After A[1]=4, the array is [0,4,7], and the max 7 is at 2.

After a[2]=3, the array is [0,4,3], and the max 4 is at 1.

After A[1]=2, the array is [0,2,3], and the max 3 is at 2.

Sample Input 1

```
3 6
2 8
2 0
1 3
1 0
2 4
0 5
```

Sample Output 1

```
2
0
1
0
2
0
```

Sample Input 2

```
5 10
0 9
0 9
4 8
0 7
4 13
4 16
4 12
4 1
1 4
2 17
```

Sample Output 2

```
0
0
0
4
4
4
4
0
0
2
```

Contest ends in 3 months

Submissions: 18
Max Score: 8
Difficulty: Medium

Rate This Challenge:

More

Java 8

```
1 import java.io.*;
```

```
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.function.*;
import java.util.regex.*;
import java.util.stream.*;
import static java.util.stream.Collectors.joining;
import static java.util.stream.Collectors.toList;



public class Solution {
    public static void main(String[] args) throws IOException {
        BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));

        String[] firstMultipleInput = bufferedReader.readLine().replaceAll("\\s+$", "").split("
");

        int M = Integer.parseInt(firstMultipleInput[0]);

        int N = Integer.parseInt(firstMultipleInput[1]);

        IntStream.range(0, N).forEach(NItr -> {
            try {
                String[] secondMultipleInput = bufferedReader.readLine().replaceAll("\\s+$",
"").split(" ");

                int i = Integer.parseInt(secondMultipleInput[0]);

                int v = Integer.parseInt(secondMultipleInput[1]);
            } catch (IOException ex) {
                throw new RuntimeException(ex);
            }
        });

        bufferedReader.close();
    }
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code