Fixed Point Removal

time limit per test

4 seconds

memory limit per test

256 megabytes

input

standard input

output

standard output

Let a1,…,ana1,…,an be an array of nn positive integers. In one operation, you can choose an index ii such that ai=iai=i, and remove aiai from the array (after the removal, the remaining parts are concatenated).

The weight of aa is defined as the maximum number of elements you can remove.

You must answer qq independent queries (x,y)(x,y): after replacing the xx first elements of aa and the yy last elements of aa by n+1n+1 (making them impossible to remove), what would be the weight of aa?

**Input**

The first line contains two integers nn and qq (1≤n,q≤3⋅1051≤n,q≤3⋅105)  — the length of the array and the number of queries.

The second line contains nn integers a1a1, a2a2, ..., anan (1≤ai≤n1≤ai≤n) — elements of the array.

The ii-th of the next qq lines contains two integers xx and yy (x,y≥0x,y≥0 and x+y<nx+y<n).

**Output**

Print qq lines, ii-th line should contain a single integer  — the answer to the ii-th query.

**Examples**

**input**

**Copy**

13 5

2 2 3 9 5 4 6 5 7 8 3 11 13

3 1

0 0

2 4

5 0

0 12

**output**

**Copy**

5

11

6

1

0

**input**

**Copy**

5 2

1 4 1 2 4

0 0

1 0

**output**

**Copy**

2

0

**Note**

Explanation of the first query:

After making first x=3x=3 and last y=1y=1 elements impossible to remove, aa becomes [×,×,×,9,5,4,6,5,7,8,3,11,×][×,×,×,9,5,4,6,5,7,8,3,11,×] (we represent 1414 as ×× for clarity).

Here is a strategy that removes 55 elements (the element removed is colored in red):

* [×,×,×,9,5,4,6,5,7,8,3,11,×][×,×,×,9,5,4,6,5,7,8,3,11,×]
* [×,×,×,9,4,6,5,7,8,3,11,×][×,×,×,9,4,6,5,7,8,3,11,×]
* [×,×,×,9,4,6,5,7,8,3,×][×,×,×,9,4,6,5,7,8,3,×]
* [×,×,×,9,4,5,7,8,3,×][×,×,×,9,4,5,7,8,3,×]
* [×,×,×,9,4,5,7,3,×][×,×,×,9,4,5,7,3,×]
* [×,×,×,9,4,5,3,×][×,×,×,9,4,5,3,×] (final state)

It is impossible to remove more than 55 elements, hence the weight is 55.