

## Problem Cards

Input file        `stdin`  
Output file      `stdout`

LITTLE CS is playing a game which involves a deck of  $N$  cards numbered from 1 to  $N$  and an initially empty board. Each card has a positive integer value written on it, so that no two cards in the deck have the same value. The game is played as follows:

1. LITTLE CS reshuffles the cards so that they are in an order of his choice. Let  $p_1, p_2, \dots, p_N$  be the order of the cards in the deck after LITTLE CS reshuffles them, where  $p_i$  is the value written on the  $i$ -th card
2. LITTLE CS repeatedly picks the first card in the deck and has two options:
  - If the card is *dominant*, he then has to place it on the board. A card is considered dominant if either the board is empty or the value written on it is at least twice as big as the value written on any card on the board.
  - If the card is not dominant, he has to throw it away and cannot use it again.
3. When the deck becomes empty, let  $S$  be the sum of the values written on the cards on the board. Then,  $S$  is the score of the game.

LITTLE CS is very competitive by nature, and desires that his score is as great as possible. In fact, he wonders what is the greatest score he could achieve. That is, what is the highest value of  $S$  that could be obtained by choosing  $p_1, p_2, \dots, p_N$  optimally. LITTLE CS also wonders what is the number of orders  $p_1, p_2, \dots, p_N$  that lead to the highest value of  $S$ .

This is too easy of a task for LITTLE CS, so instead, he decided to make it harder and give it to you. Therefore, he has chosen  $N$  cards with values  $a_1, a_2, \dots, a_N$  written on them, and  $Q$  pairs of integers  $(l, r)$  so that  $1 \leq l \leq r \leq N$ , and for each one of them, he asks you to calculate the two values he wishes to find, if he would only play the game with the cards  $a_l, a_{l+1}, \dots, a_r$ .

### Input data

The first line of input contains  $N$ , the number of cards, and  $Q$ , the number of queries you are expected to answer. The second line contains  $a_1, a_2, \dots, a_N$ , the values written on LITTLE CS' cards. Each of the following  $Q$  lines contains two integers, the  $i$ -th line containing the  $i$ -th pair  $(l_i, r_i)$  that LITTLE CS has chosen for you.

### Output data

Each of the  $Q$  lines of output must contain two space separated integers, representing the answer to LITTLE CS' queries. More specifically, the  $i$ -th line must contain the following values in the given order:

- The maximum value of  $S$  that could be obtained by playing only with the cards numbered from  $l_i$  to  $r_i$
- The number of arrays  $p_1, p_2, \dots, p_{r_i-l_i+1}$  that lead to the maximum score. Since this value could be quite large, you should output it modulo  $10^9 + 7$ .

## Implementation details

You may use the following function while implementing your solution.

```
int product(int a, int b);
```

This will return, in constant time, the value  $a \times (a + 1) \times \dots \times (b - 1) \times b \bmod 10^9 + 7$ , i.e. the product of all integers between  $a$  and  $b$ , modulo  $10^9 + 7$ . Note that the parameters  $a$  and  $b$  need to satisfy  $1 \leq a \leq b \leq 200\,000$ .

Remember to include the header `cards.h`, using the command `#include "cards.h"`! Note that you should still implement the `main` function.

## Restrictions

- $1 \leq N \leq 200\,000$
- $1 \leq Q \leq 50\,000$
- $1 \leq a_i \leq 500\,000$ , for all  $1 \leq i \leq N$ , the elements of  $a_i$  are pairwise distinct
- $1 \leq l_i \leq r_i \leq N$ , for all  $1 \leq i \leq Q$
- 50% of the points for each subtask are awarded if the maximum value of  $S$  is correct for all queries, and the other 50% are awarded if the number of arrays is correct for all queries. Please note that you still have to output the number of arrays for each query, even if they are not correct, in order to get the points for finding the values of  $S$ .

#	Points	Restrictions
1	8	$Q \leq 300, r_i - l_i \leq 7$ for all $1 \leq i \leq Q$
2	15	$N, Q \leq 300$
3	9	$N, Q \leq 5\,000$
4	10	$a_1 < a_2 < \dots < a_{N-1} < a_N$
5	19	$N \leq 5\,000$
6	21	$N, Q \leq 40\,000$
7	18	No further restrictions

## Examples

Input file	Output file
6 2 4 1 7 12 5 21 1 4 4 6	17 2 26 1
5 4 1 3 7 9 15 1 4 2 5 4 5 3 4	13 1 25 2 15 1 9 1
10 6 115 4 319 19 28 1032 516 140 9 710 1 3 2 7 4 10 5 7 1 9 6 10	438 1 1580 8 1725 10 1576 1 1729 48 1697 2

## Explanations

For each query, the numbers written on the cards that end up on the board are bold.

**First example** For the first query, the following arrays lead to the maximal value of S:

1. **1, 4, 7, 12**
2. **1, 4, 12, 7**

For the second query, the following arrays lead to the maximal value of S:

1. **5, 21, 12**