NAPC 2021 Practice Problems

July 24 & 25

No peeking until the practice starts!

Hay Taxes
Problem ID: haytaxes

Farmer John has N barns that he uses to store hay, labeled $1 \ldots N$. There are three types of operations he wishes to perform.

1. Add K bales of hay to barns L through R.
2. Remove half (rounding up if needed) of the bales of hay from each of barns L through R. So if a barn contains 9 bales of hay before this operation it will contain $9 - \lceil 9/2 \rceil = 4$ afterwards.
3. Count how many bales of hay are stored in barns L through R.

Help Farmer John perform these operations!

Input

The first line of input contains N ($1 \leq N \leq 10^5$) and Q ($1 \leq Q \leq 10^5$), the number of barns and the number of operations Farmer John will perform.

The next line contains N nonnegative integers, the ith integer being the number of bales of hay initially in barn i. Each of these numbers will be at most 10^4.

The next Q lines each start with an integer between 1 and 3, followed by two integers L and R ($1 \leq L \leq R \leq N$). The first integer corresponds to the operation as listed above. If the first integer is 1, then there will also be a fourth integer on the line, K ($1 \leq K \leq 100$).

Output

For each operation of type 3, output the number of bales of hay on its own line. The conditions guarantee that this number fits into a signed 64-bit integer.

Explanation of Sample

The answer to the first operation of type 3 is $(21 + 9) + (50 + 9) + (74 + 9) = 172$.

Sample Input 1

```
10 10
76 66 11 73 77 33 21 50 74 82
1 7 9 9
2 1 1
3 7 9
1 5 6 4
2 1 2
3 4 6
2 4 8
1 2 6 6
1 3 4 4
3 1 10
```

Sample Output 1

```
172
191
404
```

Friendly Species

Problem ID: friendlyspecies

Disclaimer: the flavor text below was initially a work of fiction. It has been edited to emphasize coincidental resemblances to actual events and locales.

Planet W has many species. These species are quite interesting: some are quite friendly with each other, while others (e.g. geese) will aggressively attack anything in sight.

To better understand these friendness level between species, planet W management has created a quantitative model. They discovered that the friendliness of two species depend on K attributes common to all species. Call them attributes $1, \ldots, K$.

Planet W management has concluded that differences among the first $K-1$ attributes increase friendliness between species. However, attribute K is different: the smaller it is, the more friendly two species are.

Specifically, let the attributes be $A[i][j]$, where $1 \leq i << n$ and $1 \leq j \leq k$. Then there are **non-negative** coefficients $C[1], \ldots, C[k]$ such that the friendliness between species i_1 and i_2 is:

$$\sum_{j=1}^{K-1} C[j] \cdot |A[i1][j] - A[i2][j]| - C[k] \cdot |A[i1][K] - A[i2][k]| \,.$$

Your job is the find the maximum friendliness between a distinct pair of species.

Input

The first line of input contains two integers N ($1 \leq N \leq 10^5$), the number of species, and K ($2 \leq K \leq 5$), the number of attributes. The second line contains the K values $C[1], \ldots C[K]$ ($0 \leq C[i] \leq 100$ for each $1 \leq i \leq K$). Finally, N lines each containing K values. The jth value on line i is $A[i][j]$ ($|A[i][j]| \leq 10^4$).

It is guaranteed that the maximum friendliness is more than 0.

Output

Output the maximum friendliness between a distinct pair of species on a line by itself.

Explanation of Sample

The friendliness between species 3 and 5 is

$$1 \cdot |0 - (-10)| + 2 \cdot |5 - (-11)| - 3 \cdot |9 - 7| = 36.$$

Sample Input 1	Sample Output 1
5 3 1 2 3 -5 3 2 -2 3 0 0 5 9 3 4 -1 -10 -11 7	36

Exhausted?
Problem ID: exhausted

There are M chairs arranged in a line. The coordinate of the i-th chair $(1 \le i \le M)$ is i.

N people of the Takahashi clan played too many games, and they are all suffering from backaches. They need to sit in chairs and rest, but they are particular about which chairs they sit in. Specifically, the i-th person wishes to sit in a chair whose coordinate is not greater than L_i, *or* not less than R_i. Naturally, only one person can sit in the same chair.

It may not be possible for all of them to sit in their preferred chairs, if nothing is done. Aoki, who cares for the health of the people of the Takahashi clan, decides to provide additional chairs so that all of them can sit in chairs at their preferred positions.

Additional chairs can be placed at arbitrary real coordinates. Find the minimum required number of additional chairs.

Input

The first line of input contains two integers N, M $(1 \le N, M \le 200\,000)$. Then N lines follow each of which contains two integers L_i, R_i $(0 \le L_i < R_i \le M + 1)$.

Output

Print the minimum required number of additional chairs.

Sample Input 1

```
4 4
0 3
2 3
1 3
3 4
```

Sample Output 1

```
0
```

Sample Input 2

```
7 6
0 7
1 5
3 6
2 7
1 6
2 6
3 7
```

Sample Output 2

```
2
```

Sample Input 3

```
3 1
1 2
1 2
1 2
```

Sample Output 3

```
2
```

Sample Input 4

```
6 6
1 6
1 6
1 5
1 5
2 6
2 6
```

Sample Output 4

```
2
```

Bubble Sort 2
Problem ID: bubblesort2

Bubble sort is an algorithm to sort a sequence. Let's say we are going to sort a sequence $A_0, A_1, \ldots, A_{N-1}$ of length N in non-decreasing order. Bubble sort swaps two adjacent numbers when they are not in the correct order. Swaps are done by repeatedly passing through the sequence. Precisely speaking, in a pass, we swap A_i and A_{i+1} if $A_i > A_{i+1}$, for $i = 0, 1, \ldots, N-2$ in this order.

It is known that any sequence can be sorted in non-decreasing order by some passes. For a sequence A, we define the number of passes by bubble sort as the number of passes needed to sort A using the above algorithm.

JOI-kun has a sequence A of length N. He is going to process Q queries of modifying values of A. In the j'th query, JOI-kun changes the value of A_{X_j} to V_j for a given index X_j and a given value V_j.

JOI-kun wants to know the number of passes by bubble sort for the sequence after processing each query. Note, these queries are cumulative meaning the array resulting from making the change after query $j-1$ is the array we modify for query j.

Input

The first line of input contains two integers N and Q ($1 \leq N, Q \leq 500\,000$) indicating the length of the array and the number of queries, respectively.

The second line contains N values $A_0, A_1, \ldots, A_{N-1}$ ($1 \leq A_i \leq 10^9$ for each $0 \leq i < N$) giving the initial contents of the array.

Then Q lines follow where the j'th line contains the values X_j ($0 \leq j \leq N-1$) and V_j ($1 \leq V_j \leq 10^9$) corresponding to the query.

Output

Output consists of Q lines, one for each query. The j'th line of output contains the number of passes by bubble sort for the sequence resulting after changing A_{X_j} to V_j.

Explanation for First Sample

Given a sequence $A = \{1, 2, 3, 4\}$ of length $N = 4$ and $Q = 2$ queries: $X = \{0, 2\}$, $V = \{3, 1\}$.

1. For the first query, the value of A_0 is changed into 3. We obtain $A = \{3, 2, 3, 4\}$.
2. For the second query, the value of A_2 is changed into 1. We obtain $A = \{3, 2, 1, 4\}$.

Bubble sort for $A = \{3, 2, 3, 4\}$:

- A is not sorted, so the first pass starts. Since $A_0 > A_1$, we swap them to get $A = \{2, 3, 3, 4\}$. Since $A_1 \leq A_2$, we don't swap them. Since $A_2 \leq A_3$, we don't swap them.
- Now A is sorted, so the bubble sort ends. Hence, the number of passes by bubble sort is 1 for $A = \{3, 2, 3, 4\}$.

Bubble sort for $A = \{3, 2, 1, 4\}$:

- A is not sorted, so the first pass starts. Since $A_0 > A_1$, we swap them to get $A = \{2, 3, 1, 4\}$. Since $A_1 > A_2$, we swap them to get $A = \{2, 1, 3, 4\}$. Since $A_2 \leq A_3$, we don't swap them.
- A is not sorted yet, so the second pass starts. Since $A_0 > A_1$, we swap them to get $A = \{1, 2, 3, 4\}$. Since $A1 \leq A2$, we don't swap them. Since $A_2 \leq A_3$, we don't swap them.
- Now A is sorted, so the bubble sort ends. Hence, then number of passes by bubble sort is 2 for $A = \{3, 2, 1, 4\}$.

Sample Input 1	Sample Output 1
4 2 1 2 3 4 0 3 2 1	1 2

Sample Input 2

```
11 12
11 4 13 6 7 3 5 12 4 10 11
8 11
4 4
6 20
0 2
7 2
3 18
5 9
0 6
8 8
9 4
0 8
6 18
```

Sample Output 2

```
5
5
5
4
6
6
6
7
7
7
7
7
```

Totem
Problem ID: totem

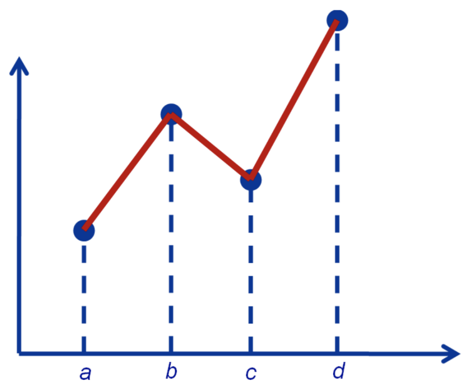
"I suppose this ought to be called history; but since all I can rely on is my memory, it lacks the rigor of history. It's not even accurate to call it the past, for the events related in these pages didn't occur in the past, aren't taking place now, and will not happen in the future..."

– prelude to Death's End (literal translation: The Grim Reaper Lives Forever)

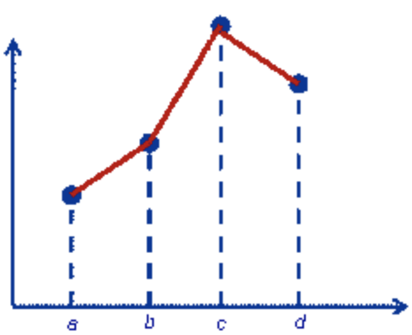
Once upon a time, there were two tribes. One worships electrical phenomena, one worships the earth. Their totems are lightning and mountain peaks respectively.

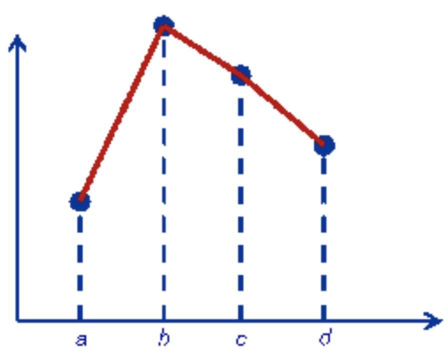
A gigantic mural was found in a cave. This mural has n dots, all with distinct horizontal and vertical positions. As only the relative positions of these dots matter, we treat them as located at $(1, y_1), (2, y_2), (3, y_3) \dots (n, y_n)$ where $y_1 \dots y_n$ is guaranteed to be a permutation of $1 \dots n$.

Archaeologists are interested in the number of totems contained in the mural. A `lightning` totem is defined as four locations $1 \leq a < b < c < d \leq n$ such that $y_a < y_c < y_b < y_d$.



The earth worshipping tribe had a split at one point. So they have two possible mountain totems: on the left is type `mountain.A`, on the right is type `mountain.B`.





Formally, four indices $1 \leq a < b < c < d \leq n$ is a `mountain.A` type totem if $y_a < y_b < y_d < y_c$, and is a `mountain.B` type totem if $y_a < y_d < y_c < y_b$.

We want to know the number of lightning totems minus the total number of mountain totems. As this number may be very large, output its residue modulo 16 777 216 (we define the remainder of -1 mod 16 777 216 to be 16 777 215).

Input

The first line of input contains a single integers N $(1 \leq N \leq 200\,000)$ indicating the number of dots. The second line contains a permutation of the integers $1, \ldots, N$. That is, the second line contains precisely N distinct integers $y_1, \ldots, y_N$ each between 1 and N. The i'th such integer is the height of the i'th dot.

Output

Sample Input 1	Sample Output 1
5 1 5 3 2 4	0

Sample Input 2	Sample Output 2
4 1 2 4 3	16777215

Ice Cream Cones
Problem ID: icecreamcones

Farmer John is opening an ice cream stand! Support the following operations:

- **ADD** K P - Farmer John is now willing to sell K more ice cream cones, each at P dollars.
- **ADDRANGE** A B - Farmer John is now willing to sell one more ice cream cone for each price P between A and B, inclusive.
- **BUYAMT** Q - Bessie has Q dollars, and buys the maximum number of cones she can, starting from cheapest to most expensive. Report how many cones Bessie buys.
- **BUYLOW** L - Bessie buys the L cheapest cones Farmer John is selling, or all of them if Farmer John is selling fewer than L of them. Report the total cost of the cones bought.
- **BUYHIGH** L - Bessie buys the L most expensive cones Farmer John is selling, or all of them if Farmer John is selling fewer than L of them. Report the total cost of the cones bought.
- **COST** L - Report the cost of the Lth cheapest cone. If there are fewer than L cones, return -1.
- **NUMCONES** - Report how many cones Farmer John is currently selling.
- **TOTALCOST** - Report the total cost of every cone that Farmer John is currently selling.

Input

The first line of input contains a single integer N ($1 \leq N \leq 3 \cdot 10^5$) indicating the number of operations to support.

Each of the next N lines contains one of the queries specified above. The values as written above will satisfy:

- $0 < K, P \leq 2 \cdot 10^6$,
- $0 < A \leq B \leq 2 \cdot 10^6$,
- $0 < L \leq 10^{12}$, and
- $0 < Q \leq 10^{18}$.

Output

For every operation that demands reporting a value, print out the desired value on a single line.

Sample Input 1	Sample Output 1
8 ADD 5 4 ADDRANGE 1 7 BUYAMT 3 BUYLOW 2 BUYHIGH 2 COST 1 NUMCONES TOTALCOST	2 7 13 4 6 25

Maximal Sequences
Problem ID: maximalsequences

The problem is simple. You are given a long sequence of integers $a_1, a_2, \ldots, a_n$. Then you are given a query consisting of a *start index* i and a subset of integers B. What is the longest consecutive subsequence of the given sequence that starts at position i and contains only integers in B?

Simple, right?

Input

The first line of input contains a single integer $1 \leq n \leq 10^5$. The second line contains n integers $a_1, \ldots, a_n$. Each integer a_j lies between 0 and $2^{31} - 1$.

The third line contains a single integer $q \geq 1$ indicating the number of queries to process. Then q lines follow, each starting with two integers $1 \leq i \leq n$ and $1 \leq m \leq 10^5$, followed by m distinct integers $b_1, \ldots, b_m$. Each integer b_j lies between 0 and $2^{31} - 1$.

Finally, you are guaranteed the sum of all values m over all queries is at most 10^5.

Output

For each query, output a single line with the length of the longest prefix of $a_i, a_{i+1}, \ldots, a_n$ that only contains integers from B.

Sample Input 1

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

Sample Output 1

```
7
2
2
0
4
```

Sample Input 2

```
10
1 2 3 4 5 6 7 8 9 10
5
1 10 1 2 3 4 5 6 7 8 9 10
7 10 1 2 3 4 5 6 7 8 9 10
5 5 1 14 7 6 5
2 6 6 3 4 2 7 5
1 1 1
```

Sample Output 2

```
10
4
3
6
1
```