Problem E. Maxim Buys an Apartment

Input file:	`standard input`
Output file:	`standard output`
Time limit:	0.5 seconds
Memory limit:	512 megabytes

Maxim wants to buy an apartment in a new house at Line Avenue of Metropolis. The house has n apartments that are numbered from 1 to n and are arranged in a row. Two apartments are adjacent if their indices differ by 1. Some of the apartments can already be inhabited, others are available for sale.

Maxim often visits his neighbors, so apartment is *good* for him if it is available for sale and there is at least one already inhabited apartment adjacent to it. Maxim knows that there are exactly k already inhabited apartments, but he doesn't know their indices yet.

Find out what could be the minimum possible and the maximum possible number of apartments that are good for Maxim.

Input

The only line of the input contains two integers: n and k ($1 \le n \le 10^9$, $0 \le k \le n$).

Output

Print the minimum possible and the maximum possible number of apartments good for Maxim.

Example

standard input	standard output
`6 3`	`1 3`

Note

In the sample test, the number of good apartments could be minimum possible if, for example, apartments with indices 1, 2 and 3 were inhabited. In this case only apartment 4 is good. The maximum possible number could be, for example, if apartments with indices 1, 3 and 5 were inhabited. In this case all other apartments: 2, 4 and 6 are good.

Scoring

There are 3 test groups. In this problem all tests are scored **separately**, each test (except sample tests) is worth 2 points.

Group	Tests	Points	Additional Constraints	Comment
			n, k	
0	1	0	–	Sample test
1	2 – 16	30	$n, k \le 10$	
2	17 – 31	30	$n, k \le 1000$	
3	32 – 51	40	–	

Problem F. Planning

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Helen works in Metropolis airport. She is responsible for creating a departure schedule. There are n flights that must depart today, the i-th of them is planned to depart at the i-th minute of the day.

Metropolis airport is the main transport hub of Metropolia, so it is difficult to keep the schedule intact. This is exactly the case today: because of technical issues, no flights were able to depart during the first k minutes of the day, so now the new departure schedule must be created.

All n scheduled flights must now depart at different minutes between $(k+1)$-th and $(k+n)$-th, inclusive. However, it's not mandatory for the flights to depart in the same order they were initially scheduled to do so — their order in the new schedule can be different. There is only one restriction: no flight is allowed to depart earlier than it was supposed to depart in the initial schedule.

Helen knows that each minute of delay of the i-th flight costs airport c_i burles. Help her find the order for flights to depart in the new schedule that minimizes the total cost for the airport.

Input

The first line contains two integers n and k ($1 \le k \le n \le 300\,000$), here n is the number of flights, and k is the number of minutes in the beginning of the day that the flights did not depart.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \le c_i \le 10^7$), here c_i is the cost of delaying the i-th flight for one minute.

Output

The first line must contain the minimum possible total cost of delaying the flights.

The second line must contain n different integers $t_1, t_2, \ldots, t_n$ ($k+1 \le t_i \le k+n$), here t_i is the minute when the i-th flight must depart. If there are several optimal schedules, print any of them.

Example

standard input	standard output
5 2 4 2 1 10 2	20 3 6 7 4 5

Note

Let us consider sample test. If Helen just moves all flights 2 minutes later preserving the order, the total cost of delaying the flights would be $(3-1)\cdot 4 + (4-2)\cdot 2 + (5-3)\cdot 1 + (6-4)\cdot 10 + (7-5)\cdot 2 = 38$ burles.

However, the better schedule is shown in the sample answer, its cost is $(3-1)\cdot 4 + (6-2)\cdot 2 + (7-3)\cdot 1 + (4-4)\cdot 10 + (5-5)\cdot 2 = 20$ burles.

Scoring

Tests for this problem are divided into three groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **previous** groups.

Group	Points	Additional Constraints	Comment
		n	
0	0	—	Sample tests
1	24	$n \leq 10$	
2	37	$n \leq 2000$	
3	39	–	

Problem G. Boredom

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Ilya is sitting in a waiting area of Metropolis airport and is bored of looking at time table that shows again and again that his plane is delayed. So he took out a sheet of paper and decided to solve some problems.

First Ilya has drawn a grid of size $n \times n$ and marked n squares on it, such that no two marked squares share the same row or the same column. He calls a rectangle on a grid with sides parallel to grid sides *beautiful* if exactly two of its corner squares are marked. There are exactly $\frac{n(n-1)}{2}$ beautiful rectangles.

Ilya has chosen q query rectangles on a grid with sides parallel to grid sides (not necessarily beautiful ones), and for each of those rectangles he wants to find its *beauty degree*. Beauty degree of a rectangle is the number of beautiful rectangles that share at least one square with the given one.

Now Ilya thinks that he might not have enough time to solve the problem till the departure of his flight. You are given the description of marked cells and the query rectangles, help Ilya find the beauty degree of each of the query rectangles.

Input

The first line of input contains two integers n and q ($2 \le n \le 200\,000$, $1 \le q \le 200\,000$) — the size of the grid and the number of query rectangles.

The second line contains n integers $p_1, p_2, \ldots, p_n$, separated by spaces ($1 \le p_i \le n$, all p_i are different), they specify grid squares marked by Ilya: in column i he has marked a square at row p_i, rows are numbered from 1 to n, bottom to top, columns are numbered from 1 to n, left to right.

The following q lines describe query rectangles. Each rectangle is described by four integers: l, d, r, u ($1 \le l \le r \le n$, $1 \le d \le u \le n$), here l and r are the leftmost and the rightmost columns of the rectangle, d and u the bottommost and the topmost rows of the rectangle.

Output

For each query rectangle output its beauty degree on a separate line.

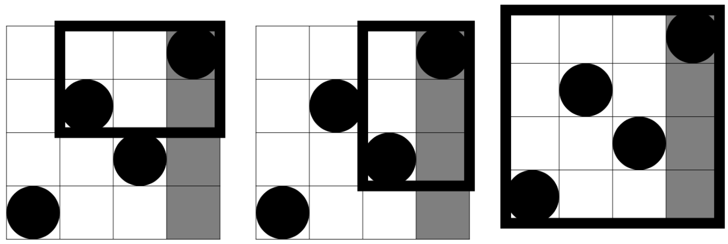
Examples

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

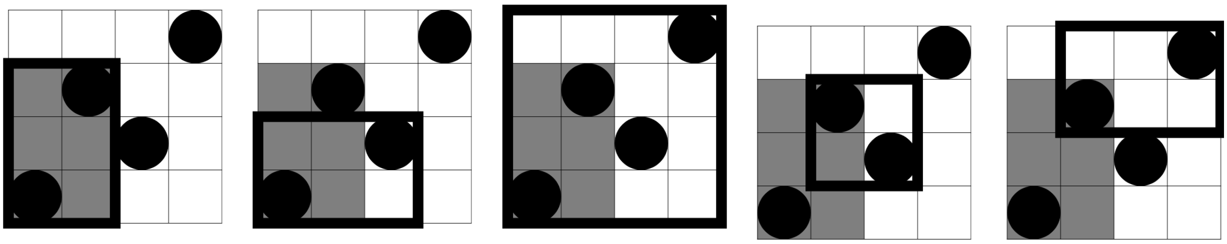
Note

The first sample test has one beautiful rectangle that occupies the whole grid, therefore the answer to any query is 1.

In the second sample test the first query rectangle intersects 3 beautiful rectangles, as shown on the picture below:



There are 5 beautiful rectangles that intersect the second query rectangle, as shown on the following picture:



Scoring

Tests for this problem are divided into five groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in all of the **previous** groups.

Group	Points	Additional Contraints	Comment
		n, q	
0	0	–	Sample tests
1	12	$n, q \leq 10$	–
2	21	$n, q \leq 100$	–
3	26	$n, q \leq 5000$	–
4	18	$n, q \leq 50\,000$	–
5	23	–	–

Problem H. Offline Pirates

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

While everybody was discussing piracy in online, serious offline piracy stroke back!

Your island looks like a convex polygon. It was surrounded by ships of offline pirates. Island is large enough to consider ships as points.

Define power of a group of offline pirate as the number of pairs of ships that see each other. Ships see each other if there is no point of their connecting segment that lies strictly inside island. In particular, if the segment connecting ships contains a different ship or passes through the island border (without containing its interior points), then ships are still able to see each other.

Find out the power of the pirate group.

Input

First line of input contains two integers k and n ($3 \le k \le 100\,000$, $1 \le n \le 100\,000$), the number of vertices of the island polygon and the number of ships.

Each of the following k lines contains two integers px_i, py_i ($-10^9 \le px_i, py_i \le 10^9$), the coordinates of polygon vertices. Vertices follow in counter-clockwise order, no three consecutive vertices lie on the same line. It is guaranteed that polygon is convex.

Each of the following n lines contains two integers sx_i, sy_i ($-10^9 \le sx_i, sy_i \le 10^9$), ship coordinates. It is guaranteed that no ship lies neither inside polygon nor on its border.

Output

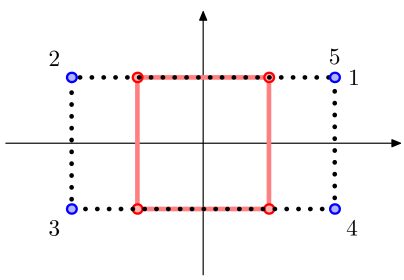
Output the power of the pirate group.

Example

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

Note

Picture for the sample case is given below.



Scoring

Tests for this problem are divided into six groups. For each of the groups you earn points only if your solution passes all tests in this group and all tests in some of the previous groups.

Group	Points	Additional constraints		Required groups	Comment
		k	n		
0	0	–	–	–	Sample tests
1	16	$k \le 10$	$n \le 2000$	0	
2	16	$k \le 1000$	$n \le 2000$	0, 1	
3	16	$k \le 200$	$n \le 7000$	0, 1	
4	16	$k \le 10$	$n \le 100\,000$	0, 1	
5	36	–	–	0 – 4	