Problem A. Score analysis

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

MSAI teaches continue to analyse results of this year courses. And again, they can't do without your help!

This time number of students was $0 \leqslant N \leqslant 100000$, and teachers want to check if courses were too easy or too difficult. To do that they need to check number of students whose average score is within given range: $l \leqslant score \leqslant r$.

Please, help the teachers and write the program which can process such requests fast. The number of requests teachers prepared for you is $0 \leqslant M \leqslant 100000$.

Input

First line contains two integer numbers divided by space character: $0 \leqslant N, M \leqslant 100000$ — number of students and requests respectively. Second line contains N numbers divided by space character: scores of students. $0 \leqslant s_i \leqslant 10^9$. Each of the following M lines contains two integer numbers: l_j, r_j, divided by space character: $0 \leqslant l_j \leqslant r_j \leqslant 10^9$

Output

For each request (l_j, r_j) print one integer number on separate line: number of students who got score within requested range: $|\{i : l_j \leqslant s_i \leqslant r_j\}|$.

Examples

`standard input`	`standard output`

`6 6`
`1 1 2 2 2 3`
`1 1`
`2 2`
`3 3`
`1 2`
`2 3`
`1 3`
	`2`
	`3`
	`1`
	`5`
	`4`
	`6`

`2 3`
`1 1`
`0 0`
`0 1000000000`
`5 10`
	`0`
	`2`
	`0`

`2 3`
`1 1000000000`
`0 10`
`10 1000000000`
`0 1000000000`
	`1`
	`1`
	`2`

Note

You can use `sorted()` and `list.sort()` in this week home assignments.

Problem B. Sigma-trimpasation 1

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

After graduating from MIPT you've decided to apply your knowledge in AI to Physics. You've become a world-wide famous physicist due to your deep knowledge of AI which you get on MSAI program. And now you are working on a innovational quantum process research: sigma-trimpazation*. Your research is definitely going to change the world!

Your experiment is in progress. Quantum sensors are installed and send you data permanently. You've found a regularity in this data. And now you are going to analyse it using python. What you need – is to get this data in sorted order.

Your process is initialized with three parameters: $0 \leqslant N \leqslant 10^7$, $0 < M \leqslant 10^4$, $0 < q_0 < 2^{31}$. The process is the following: your data sequence x is generated from quantum data sequence q. You are given q_0 value and next items are generated using the following rule** (this process is already implemented for you):

$$q_i = (q_{i-1} * 7^5) \ mod \ (2^{31} - 1)$$

The data you need to analyze is calculated from quantum data: $x_i = (q_i \ mod \ M) - M//2$

Let's denote sorted data sequence as $x_s = sorted(x)$. You need to calculate the following value:

$$y = \sum_{i=1}^{N} (i * x_s[i-1]) = 1 * x_s[0] + 2 * x_s[1] + 3 * x_s[2] + ... + N * x_s[N-1]$$

You've tried to use built-it python sorting algorithm but it takes too much time (your code is here: `https://bit.ly/34vIxb2`). You need to implement something faster. Try to optimize it somehow. No time to obtain TL verdict! Research must go on!

Input

The only line in input file contains 3 integer numbers divided by space character and followed by line break: $0 \leqslant N \leqslant 10^7$, $0 < M \leqslant 10^4$, $0 < q_0 < 2^{31}$ — parameters of your process.

Output

Print the only integer number — y value for your process.

Examples

standard input	standard output
3 5 1	5 # array generated: # [-1, 0, 5] # result: # -1 * 1 + 0 * 2 + 5 * 2 = 5
5 10 7	29 # array generated: # [2, 4, -2, 1, 0] # sorted: # [-2, 0, 1, 2, 4] # result: # -2 * 1 + 0 * 2 + 1 * 3 + # 2 * 4 + 4 * 5 = 29
10000000 10000 1	83287854395709985

Note

*Trimpazation — joking name of space warp process in book of famous russian sci-fiction writers — the Strugatsky brothers (original: тирьямпампация).

A. and B. Strugatsky (1962). *Noon: XXII century* (translated by Patrick L.McGuire). ISBN:0-02-615150-2.

** This process of «quantum» data generation is a Lehmer pseudo-random numbers generator: `https://en.wikipedia.org/wiki/Lehmer_random_number_generator`, and $q[0]$ — it's random seed. This generator is used here not to slow your program down with reading a lot of data from text file (that will be really slow).

Problem C. No cheating

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

MSAI teachers are organizing an exam in another university and they are worried about cheating possibility. Of course, this exam is not in MSAI and not in MIPT, because MIPT students do not cheat. Usually :)

So, teachers decided to prepare sitting places for examinees on a straight line, to be able to see them all. Also, to minimize possibility of using each other's solutions, teachers want to maximize minimum distance between students. Help them, please!

Classroom has $2 \leqslant N \leqslant 10000$ sitting places, which are situated on a straight line. Their coordinates are $0 \leqslant x_i \leqslant 10^9$. You need to assign M students to seats (choose M seats: $x_{k_1}, x_{k_2}, ..., x_{k_M}$) so that minimum distance between students is as much as possible:

$$\min_{i \neq j} |x_{k_i} - x_{k_j}| \to \max$$

Write a program which will do that.

Input

First line contains 2 integer numbers N, M divided by space character: $2 \leqslant N \leqslant 10000$, $2 \leqslant M \leqslant N$ — number of sitting places and students respectively.

Next line contains N integer numbers $0 \leqslant x_i \leqslant 10^9$ — coordinates of sitting places.

Output

Print single integer number — desired maximized minimum distance: $\min\limits_{i \neq j} |x_{k_i} - x_{k_j}|$

Examples

`standard input`	`standard output`

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

`7 3`
`8 9 4 7 12 15 1`
	`7`

`6 4`
`1000000000 0 1 10 11 100`
	`11`