

Orderbook Omnishamble

time limit per test: 3 seconds (3 seconds for Python)

memory limit per test: 256 megabytes

input: standard input

output: standard output

points: 100 · problem code: orderbookomni · author: de

The panic button has been hit! Your firm is currently swiftly retreating from the market after the orderbook glitched and you've been called to help with the recovery. You are an analyst estimating losses to claim, and to solve this issue, you are given:

- A sorted (ascending) array of integers n , representing the spread of last known orders.
- A zero or positive integer k , representing your estimated capital.
- A positive integer c , representing a range of risk values.

These represent the spread of the last known orders, estimated capital, and identify a range of risk values respectively. This risk value range is represented by the integer c , representing that the range encompasses the $\frac{n}{c}$ -th smallest number to the $\frac{n}{c}$ -th highest number in the array n . The integer k , your estimated capital, shall be used for operations on the values in the report you are writing up.

Each operation will cost 1 unit of your theoretical capital, but in exchange, you may increment or decrement any value in n by 1 (this will alter the risk range!). Your task is to **maximize the risk range**, namely, maximize the difference between the $\frac{n}{c}$ -th highest number and the $\frac{n}{c}$ -th smallest number in the array n , making only k operations.

Your company has told you, luckily, that the elements in the array n will always be a multiple of c , but that is all.

Good luck analyst, your company relies on you.

Input

- The first line contains integers $|n|$ (length of array), k (capital), and c (range).
- The second line contains n integers, representing the sorted array.

Output

- A single integer containing the maximum risk range achievable with up to k operations.

Constraints

- $2 \leq c \leq n$
- $n \bmod c = 0$
- $2 \leq n \leq 10^5$
- $0 \leq k \leq 10^9$
- $-10^5 \leq n_i \leq 10^5$

Example 1

Input

```
12 1 3
-8 -8 7 7 7 7 7 7 7 8 8 8
```

Output

```
1
```

Example 2

Input

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

Output

1

Example 3

Input

12 5 12
1 1 1 1 1 1 1 1 1 1 1 1

Output

5

Example 4

Input

6 0 3
5 7 9 12 13 15

Output

6

Example 5

Input

9 6 3
-5 1 1 4 4 4 4 6 7

Output

7

Example 6

Input

12 5 3
-7 -6 -3 -2 -2 -2 -2 -2 -2 -2 -2 -1

Output

3