

## Restaurant Seating

You own a restaurant, and you want your customers to be as happy as possible. You've learned over the years that customers really don't like sitting very close to each other. They prefer if others cannot hear their conversations. This however, is a pretty hard problem, so you've decided to construct an algorithm that can figure out where to seat your patrons to maximize the distance between them.

Your restaurant is in an old building, and thus has a peculiar shape. The restaurant is very long and not very wide (it was the cheapest space available for lease!). Because of this your tables are all in a horizontal line down your restaurant, but the distances between the tables vary. Given a list of the positions of the tables in your restaurant, and the number of patrons that wish to be seated, return the maximum distance between any two of the patrons after being seated in the optimal arrangement.



### Input

The input file will begin with one line containing  $n \leq 10^6$  and  $p \leq 10^6$ , the number of tables in your restaurant and the number of patrons to sit respectively. The following  $n$  lines will each contain a single integer  $l_i \leq 10^8$  describing the integer position of table  $i$ . These positions will be sorted in increasing order. All positions will be unique.

### Output

Output the largest possible value of the minimum distance between patrons after being seated in the optimal seating arrangement.

### Sample Input

```
5 3
1
2
4
8
9
```

### Sample Output

```
3
```