

You will be given a list of integers, *arr*, and a single integer *k*. You must create an array of length *k* from elements of *arr* such that its *unfairness* is minimized. Call that array *arr'*. Unfairness of an array is calculated as

$$\max(arr') - \min(arr')$$

Where:

- *max* denotes the largest integer in *arr'*.
- *min* denotes the smallest integer in *arr'*.

Example

arr = [1, 4, 7, 2]

k = 2

Pick any two elements, say *arr'* = [4, 7].

unfairness = *max*(4, 7) - *min*(4, 7) = 7 - 4 = 3

Testing for all pairs, the solution [1, 2] provides the minimum unfairness.

Note: Integers in *arr* may not be unique.

Function Description

Complete the *maxMin* function in the editor below.

maxMin has the following parameter(s):

- *int k*: the number of elements to select
- *int arr[n]*:: an array of integers

Returns

- *int*: the minimum possible *unfairness*

Input Format

The first line contains an integer *n*, the number of elements in array *arr*.

The second line contains an integer *k*.

Each of the next *n* lines contains an integer *arr[i]* where $0 \leq i < n$.

Constraints

$$2 \leq n \leq 10^5$$

$$2 \leq k \leq n$$

$$0 \leq arr[i] \leq 10^9$$

Sample Input 0

```
7
3
10
100
300
200
1000
20
30
```

Sample Output 0

```
20
```

Explanation 0

Here $k = 3$; selecting the 3 integers 10, 20, 30, unfairness equals

```
max(10,20,30) - min(10,20,30) = 30 - 10 = 20
```

Sample Input 1

```
10
4
1
2
3
4
10
20
30
40
100
200
```

Sample Output 1

```
3
```

Explanation 1

Here $k = 4$; selecting the 4 integers 1, 2, 3, 4, unfairness equals

```
max(1,2,3,4) - min(1,2,3,4) = 4 - 1 = 3
```

Sample Input 2

```
5
2
1
2
1
```

2
1

Sample Output 2

0

Explanation 2

Here $k = 2$. $arr' = [2, 2]$ or $arr' = [1, 1]$ give the minimum unfairness of 0.