

Sort

Information

| Time Limit | Memory Limit | Data Amount | Problem Type |
|-------------------------|--------------|-------------|--------------|
| 1000ms(1200ms for Java) | 128MiB | 10 | Tradition |

Description

Peter cleverly designed an integer arrays $A : \{a_1, a_2, \dots, a_n\}$

He wants you to implement the sorting algorithm yourself.

Input

The first line is n .

The second line is n integers, $a_1 \ a_2 \ \dots \ a_n$.

Output

The sorted array.

Sample Test Data

```
5
4 3 5 1 2
<|==|>
1 2 3 4 5
```

Data Limit

For 20% cases: $1 \leq n \leq 2000$

For 70% cases: $1 \leq n \leq 100000$, $-10^9 \leq a_i \leq 10^9$

For **other** 30% cases: $1 \leq n \leq 5000000$, $-10^5 \leq a_i \leq 10^5$

For $n > 100000$ cases, the output can be huge and if A is your sorted array, you only need to output $A[0] \ A[100] \ A[200] \ \dots$ (i.e. $A[k]$ if $k \% 100 == 0$).

Attention

You should implement sorting algorithm yourself.

Do not use libraries related to sorting, such as `Array.sort()`, `TreeMap`, `PriorityQueue`, `sort()`, `map`, `priority_queue`, etc.

The input can be huge, and you can reduce input time by using fast input.

C++ fast input:

```

int read(int x = 0, int f = 1, char ch = getchar()) {
    while(ch < '0' || ch > '9') {
        if(ch == '-') f = -1;
        ch = getchar();
    }
    while(ch >= '0' && ch <= '9') x = x * 10 + ch - '0', ch = getchar();
    return x * f;
}

//in main
int main() {
    int a;
    a = read();
}

```

Java fast I/O: <https://paste.ubuntu.com/p/6ybMcVxz5/>

Lazy Bob

Information

| Time Limit | Memory Limit | Data Amount | Problem Type |
|-------------------------|-------------------------|-------------|--------------|
| 1000ms(1200ms for Java) | 128MiB(256MiB for Java) | 2 | Tradition |

Description

Peter generates an integer array of length n in random order. Bob is so lazy that he doesn't want to sort this numbers, although he has just learned sorting. But he still wants to know the k -th biggest number, can you help him? (Hint: You can find inspiration from an efficient sorting algorithm.)

Input

The first line is n, k .

The second line is n **distinct** integers, representing the array.

Output

The k -th biggest number.

Sample Test Data

```

5 2
4 3 5 1 2
<|==|>
4

```

Data Limit

For 20% cases: $1 \leq n \leq 100000$

For 100% cases: $1 \leq n \leq 5000000$, $1 \leq k \leq n$, The absolute value of integers in the array is not exceeding 10^9 .

The integers in the array are all **distinct**.

Attention

Do not use libraries related to sorting, such as `Array.sort()`, `TreeMap`, `PriorityQueue`, `sort()`, `map`, `priority_queue`, etc.

The input can be huge, and you can reduce input time by using fast input.

C++ fast input:

```
int read(int x = 0, int f = 1, char ch = getchar()) {
    while(ch < '0' || ch > '9') {
        if(ch == '-') f = -1;
        ch = getchar();
    }
    while(ch >= '0' && ch <= '9') x = x * 10 + ch - '0', ch = getchar();
    return x * f;
}
//in main
int main() {
    int a;
    a = read();
}
```

Java fast I/O: <https://paste.ubuntu.com/p/6ybMcVxz5/>

Two Rectangles

Information

| Time Limit | Memory Limit | Data Amount | Problem Type |
|------------|--------------|-------------|--------------|
| 1000ms | 128MiB | 10 | Tradition |

Description

Alice drew n points in two-dimensional field. She wants you to draw 2 rectangles whose sides are parallel to the x and y axes to enclose all of the points. The two rectangles are not allowed to overlap. The side of length 0 is legal. Please find the minimum total area of two rectangles that meet the condition.

Input

The first line is n .

The next n lines is the coordinates of each point, represented by 2 integers.

Output

The minimum total area of two rectangles that meet the condition.

Sample Test Data

```
4
0 0
1 1
9 9
10 10
<|==|>
2
```

```
6
7 2
5 2
10 6
5 2
4 6
4 5
<|==|>
6
```

Tips:

For sample test 2: one rectangle is $1 * 6$ and the other is $0 * 2$.

Data Limit

For 30% cases: $3 \leq n \leq 1000$

For 100% cases: $3 \leq n \leq 100000$

The coordinates are positive and not exceeding 10^9

Ranking Movies

Information

| Time Limit | Memory Limit | Data Amount | Problem Type |
|------------|--------------|-------------|--------------|
| 1000ms | 128MiB | 10 | Tradition |

Description

You and Peter are talking about n movies, which are represented by integers $1 \sim n$. You have made a ranking list for the movies according to your preference. Now, Peter tells you his ranking list. You want to know how similar your and Peter's tastes are. For 2 movies i, j , if you and Peter both rank movie i before movie j , You will get 1 similarity. Please output the total similarity.

Input

The first line is n .

The second line is your ranking list.

The third line is Peter's ranking list.

Output

The total similarity.

Sample Test Data

```
5
2 3 4 1 5
4 2 1 3 5
<|==|>
7
```

Tips

You and Peter both rank 1, 2, 3, 4 before 5, you get 4 similarity.

You and Peter both rank 2, 4 before 1, you get 2 similarity.

You and Peter both rank 2 before 3, you get 1 similarity.

The total similarity is 7.

Data Limit

For 30% cases: $1 \leq n \leq 1000$

For 100% cases: $1 \leq n \leq 100000$

Sort II

Information

| Time Limit | Memory Limit | Data Amount | Problem Type |
|------------|--------------|-------------|--------------|
| 1000ms | 128MiB | 10 | Tradition |

Description

Peter wants you to sort an array by swapping arbitrary two numbers multiple times, and each time the cost is the sum of the two numbers.

Peter wants to know the minimum cost to sort the array.

Input

The first line is n .

The next line is n **distinct** integers, representing the array.

Output

The minimum cost to sort the array by Peter's ruler.

Sample Test Data

```
5
4 3 5 1 2
<|==|>
17
```

```
8
8 3 5 1 2 11 10 9
<|==|>
42
```

Tips

For sample test 1:

4 3 2 1 5

$cost = 2 + 5 = 7$

4 2 3 1 5

$cost = 2 + 3 = 5$

1 2 3 4 5

$cost = 1 + 4 = 5$

Total cost is $5 + 5 + 7 = 17$.

Data Limit

For 30% cases: $1 \leq n \leq 10$

For 100% cases: $1 \leq n \leq 100000$

The integers in the array are positive and not exceeding 10^9 .

The integers in the array are all **distinct**.

