BigNumber

You are given an array of integers a of length n.

In one operation you:

- Choose an index i such that $1 \le i \le n-1$ and $a_i \le a_{i+1}$.
- Increase a_i by 1.

Find the maximum possible value of $\max(a_1, a_2, \ldots a_n)$ that you can get after performing this operation at most k times.

Input

Each test contains multiple test cases. The first line of input contains a single integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains two integers n and k ($2 \le n \le 1000$, $1 \le k \le 10^8$) — the length of the array a and the maximum number of operations that can be performed.

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^8$) — the elements of the array a.

It is guaranteed that the sum of n over all test cases does not exceed 1000.

Output

For each test case output a single integer — the maximum possible maximum of the array after performing at most k operations.

Example

input

```
6
3 4
1 3 3
5 6
1 3 4 5 1
4 13
1 1 3 179
5 3
4 3 2 2 2
5 6
6 5 4 1 5
2 17
3 5
```

output

```
4
7
179
5
7
6
```

Note

In the first test case, one possible optimal sequence of operations is: $[1,3,3] \to [2,3,3] \to [2,4,3] \to [3,4,3] \to [4,4,3]$.

In the second test case, one possible optimal sequence of operations is: $[1,3,4,5,1] \to [1,4,4,5,1] \to [1,5,4,5,1] \to [1,5,5,5,1] \to [1,5,6,5,1] \to [1,6,6,5,1] \to [1,7,6,5,1]$