

Minimize the maximum difference between adjacent elements in an array

Problem Statement

You are given a non-decreasing array and an integer K . You need to remove exactly K integers from the given array such that the maximum difference between adjacent elements is minimum.

For Example:

If the given array is: `[2 6 7 7 10]` and $K = 2$. We need to remove $A[0] = 2$ and $A[4] = 10$, then the resultant array would become `[6 7 7]`, where the difference between adjacent pairs are $\{1, 0\}$. Thus our answer would be 1. You can see that there would not be any better answer than 1 for this array

Detailed explanation (Input/output format, Notes, Images)



Input Format:

The first line of input contains a single integer T , representing the number of test cases or queries to be run.

Then the T test cases follow.

The first line of each test case contains two space-separated integers N and K representing the length of the array and the number of integers to be removed.

The second line of each test case contains N space-separated integers denoting the elements of the given array.

Output Format:

For each test case, print the maximum difference between adjacent elements is minimum after K integers are removed, in a separate line.

Constraints:

$1 \leq T \leq 100$
 $3 \leq N \leq 1000$
 $1 \leq A_i \leq 10^6$
 $0 \leq K \leq N - 2$

Time Limit : 1 sec

Sample Input 1:

3

5 2

2 6 7 7 10

3 1

4 6 6

4 0

3 6 6 7

Sample Output 1:

1

0

3

Explanation Of Input 1:

The first test case has already been explained in the problem statement.

For the second test case, the given array is: [4 6 6] and $K = 1$. We remove $A[0] = 4$, then the resultant array would become [6 6]. So the answer would be 0.

For the third test case, the given array is: [3 6 6 7] and $K = 0$. We cannot remove any number. The array remains the same. So the answer becomes 3.

Sample Input 2:

3

9 6

3 3 4 6 7 10 10 12 15

4 0

1 1 3 3

9 7

1 1 2 5 7 10 13 16 17

Sample Output 2

1

2

0