Assignment 4

Heaps and Sorting
Data Structures and Algorithms
Due Date: 6 June, 2022

1 Maximum Pitch

You are given an array of n elements $a_1, a_2, \ldots, a_n$.

We define a function $f(x) :=$ minimum distance between any two consecutive occurrences of x in an array. By distance we mean the number of elements present between those two numbers.
Note :- If x occurs only one time in the array, you can safely assume $f(x)$ to be n.

Suppose there is an array $[1, 5, 6, 1, 2, 1, 3]$:- then $f(1) = \min(2, 1) = 1$. Between first and second occurrences of 1, we have 2 elements, and between second and third we have 1 element. So minimum of both will be the value of $f(1)$.

You can perform none or several operations on the array. In one operation you can swap any two elements a_i and a_j in the array such that $a_i \neq a_j$.

Pitch of an array is defined as the minimum of $f(x)$ over all the unique elements present in the array.

You have to return the **maximum** pitch achievable for the array, after you have made all the necessary swaps.

1.1 Input

The first line contains a single integer t $(1 \leq t \leq 100)$ — the number of testcases.
The first line of each testcase contains a single integer n $(1 \leq n \leq 10^4)$ — the number of elements in the array.
The second line of each test case contains n space-separated integers $a_1, a_2, \ldots, a_n$, $(1 \leq a_i \leq 10^9)$ — the elements of the array.

1.2 Output

Print a single number p, denoting the maximum pitch achievable in the array.

1.3 Example

Input	Output
4	
7 1 7 1 6 4 4 6	3
8 1 1 4 6 4 6 4 7	2
3 3 3 3	0
6 2 5 2 3 1 4	4

1.4 Note

For the first test case, we can arrange the array (after making some swaps) as follows :-
$[1, 6, 4, 7, 1, 6, 4]$.

Here $f(1) = 3, f(4) = 3, f(6) = 3$ and $f(7) = 7$. Now minimum of $f(1), f(4), f(6)$ and $f(7)$ is 3. So 3 is the answer.

2 Lego Towers

Prodigal Penelope is playing with lego blocks, and her mom decides to give her a task. She stacks the legos and forms a lineup of N lego towers of different heights. Then she tells Penelope that she has to sort the lineup of lego towers as follows.

Penelope is given a swapper tower of height K at the start. Penelope can choose any tower from the N towers, say the i^{th} tower, and check if its length is greater than K. If yes, Penelope could swap the i^{th} tower with the swapper tower.

Is it possible for Penelope to sort the lego tower lineup? If yes, what is the minimum number of swaps required to perform this sorting?

2.1 Input

The first line of input contains two space-separated integers, N ($1 \leq N \leq 1000$) and K ($1 \leq K \leq 1000$) — the number of lego towers and the height of the initial swapper tower.
The second line contains N space-separated integers $H_1, H_2, \ldots, H_N$, ($1 \leq H_i \leq 1000$) — the heights of the lego towers.

2.2 Output

Print one integer representing the minimum number of swappings Penelope must perform to make the lego tower lineup sorted. If it is not possible to sort the lego tower lineup, print -1.

2.3 Examples

2.3.1 Example 1

Input	Output
5 25 67 128 100 300 128	3

In first move, swap the swapper tower of height 25 with the first tower with height 67. Then swap the swapper tower of height 67 with the second tower of height 128. Then finally, swap the fourth tower with height 300 with the swapper tower of height 128. Thus, 3 swaps are required to sort the lego tower lineup.

2.3.2 Example 2

Input	Output
2 100 101 99	-1

3 Maximum Cost

You are given an array of n elements $a_1, a_2, \dots, a_n$.

You are asked to divide the whole array into k continuous segments, and each element must be a part of exactly one segment. Segments are numbered from left to right and from 1 to k.

Note : Every segment must be non-empty i.e. every segment must have atleast one element present.

Define another array $b_1, b_2, \dots, b_n$ where b_i is the number of the segment a_i is contained in. The cost of an array is defined as

$$\text{cost} = \sum_{i=1}^{n} a_i \cdot b_i$$

You have to return the maximum cost that we can get after optimally dividing the array into k segments.

3.1 Input

The first line contains two integers n $(1 \leq n \leq 10^4)$ and k $(1 \leq k \leq n)$ — the number of elements in the array and the number of segments to split the array into.
The second line of the test case contains n space-separated integers $a_1, a_2, \dots, a_n$, $(-10^9 \leq a_i \leq 10^9)$ — the elements of the array.

3.2 Output

Print the maximum cost you can obtain by dividing the array a into k continuous segments.

3.3 Example

Input	Output
5 2 -1 -2 5 -4 8	15
7 6 -3 0 -1 -2 -2 -4 -1	-45
4 1 3 -1 6 0	8

3.4 Note

For the first test case, we can divide the array into two subsegments like $[\{-1, -2\}, \{5, -4, 8\}]$. Thus the array b will be $[1, 1, 2, 2, 2]$. And the cost is $-1 \times 1 + (-2) \times 1 + 5 \times 2 + (-4) \times 2 + 8 \times 2 = 15$. This is the maximum cost achievable.

4 Para Commando Operation

You are a para commando on a covert operation. You have landed on a secret Island in the Indian Ocean and are supposed to disintegrate a long train made by the Aliens. The train contains L connected bogies. Your task is to disintegrate the train into D different parts, where each part could have one or more connected bogies.

On each disintegration step, you can choose a connected part of the train and divide it into two parts. For example, if you have a part with 10 connected bogies, you can divide that part into two different parts where the sum of the number of bogies in each part would be 10. However, the cost for the disintegration step is the number of bogies in the initial connected part. So in the example mentioned, the cost for disintegration is 10.

Additionally, you must disintegrate the train into components of specified lengths, which the government provided to you in a concealed document.

Before starting the mission, you need to determine the minimum cost for disintegrating the train.

4.1 Input

The first input line has two space-separated integers, L $(1 \leq L \leq 10^9)$ and D $(1 \leq D \leq 3 \times 10^5)$ — the length of the train and the number of parts to disintegrate the train into.
The second line contains D space-separated integers, $l_1, l_2, \ldots, l_d$ $\left(\sum_{i=1}^{D} l_i = L\right)$ — the required length of components after disintegrating the train.

4.2 Output

Print only one integer representing the minimum cost for disintegrating the train.

4.3 Example

Input	Output
14 3 4 5 5	23

The train is first divided into two parts of each lengths 9 and 5 with a cost of 14. Then the part with length 9 is again divided into two parts of lengths 4 and 5 with a cost of 14. Thus the total cost for disintegrating the train is $14 + 9 = 23$.

5 PEC Sports Purchase

Mr. Lakar is the head of PEC at IIIT Hyderabad. He is at a sports shop, and there are different sections to choose items to buy. The Dean of Sports has given him the responsibility of making k different orders, where he needs to take exactly one item from each section.

Interestingly, there are k sections in the sports shop, and each section has k items available. For $k = 3$, let there be three sections as follows.

- Section 1(FB): Football 1, Football 2, Football 3
- Section 2(CB): Cricket Bat 1, Cricket Bat 2, Cricket Bat 3
- Section 3(BB): Basket Ball 1, Basket Ball 2, Basket Ball 3

Then Mr. Lakar will be able to make 27 orders such as $(FB1, CB1, BB1)$, $(FB1, CB1, BB2)$, $(FB1, CB1, BB3)$, etc. But he cannot make orders like $(FB1, CB1, CB2)$, $(FB3, FB2, FB1)$; because two items are being taken from the same section.

Unfortunately for Mr. Lakar, he has to minimize the total cost for a order and report the amount required for the k cheapest orders to the Dean of Sports. But fortunately for you, he calls you to the sports shop and gives you this task. Once you solve the problem, Mr. Lakar will reward you with two PT attendance, and he might even tell TAs to give you some grace scores in DSA!

5.1 Input

The first line of the input will contain a single integer T $(1 \leq T \leq 100)$ — the number of test cases.
The first line of each test case will contain a single integer k $(1 \leq k \leq 1000)$ — the number of sections in the sports shop.
The following k lines in that test case will contain k integers each. The j^{th} integer on the i^{th} line, a_{ij} $(1 \leq a_{ij} \leq 10^6)$ indicates the price of the j^{th} item in section i.

5.2 Output

A single line of output with k space-separated integers showing the price of the k cheapest orders in non-decreasing order.

5.3 Example

Input	Output
2 3 1 8 5 9 2 5 10 7 6 2 1 1 1 2	9 10 12 2 2