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 all two consecutive occurrences of x in an array. By distance we mean the number of elements present between those two numbers

Suppose there is an array 1,5,6,1,2,1,3 :- then f(1) = min(2,1) = 2. 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/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 - the number of elements in the array.

The second line of the test case contains $n(1 \leq n \leq 10^4)$ 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

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.

How many swaps will Penelope have to do to sort the lego tower lineup? Is it impossible?

2.1 Input

The first line of input contains two space-separated integers, N and K.

The second line contains N space-separated integers $H_1, H_2, ..., H_N$, representing 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's impossible to make the lego tower lineup sorted, print -1.

2.3 Constraints

$1 \leq N \leq 1000$
$1 \leq K \leq 1000$
$1 \leq H_i \leq 1000$

2.4 Examples

2.4.1 Example 1

Input	Output
5 25 67 128 100 300 128	3

In first move, swap 25 with 67. Then swap 67 with 128. Then finally, swap the tower with height 300 with the swapper tower.

2.4.2 Example 2

Input	Output
2 100 101 99	-1

3 Maximum Cost

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

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 $b_1, b_2, \ldots, b_n$ where b_i is the segment number of a_i it contains in.

Cost of an array is defined as

$$cost = \sum_{n=1}^{n} a_i * b_i$$

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 a single integer *nandk* - the number of testcases.

The first line of each testcase contains a single integer n - the number of elements in the array.

The second line of the test case contains $n(1 \leq n \leq 10^4)$ space-separated integers $a_1, a_2, \ldots, a_n$, $(1 \leq a_i \leq 10^9)$ 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

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 ten 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 ten.

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 and D. The second line contains D space-separated integers, l_1, l_2, ...l_d: where l_is represent the required length of components after disintegrating the train.

4.2 Output

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

4.3 Constraints

$1 \leq L \leq 10^9$
$1 \leq D \leq 3 * 10^5$
$\sum_{i=1}^{D} l_i = L$

4.4 Example

Input	Output
14 3 4 5 5	23

The train is first divided into two parts, lengths 9 and 5. Then the part with length nine is again divided into two parts of lengths 4 and 5. 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 precisely one item from each section. But all k items of any two orders should not match exactly.

Interestingly, there are precisely k sections in the sports shop, and each section has k items available. Let's say there are 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

Mr. Lakar will be able to make three orders such as (FB1, CB1, BB1), (FB2, CB1, BB1), (FB3, CB3, BB3). But he cannot make orders like (FB1, CB1, BB1), (FB3, CB3, BB3), (FB3, CB3, BB3); because here, the second and third orders are the same.

Unfortunately, Mr. Lakar has to minimize the total purchase cost and report the amount of money to the Dean of Sports. But fortunately for you, he calls you to the sports shop and gives you the task of finding the total purchase cost. 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!

You will be given the price of each item in all k sections of the shop. You are required to find the cost of the cheapest k purchases.

5.1 Input

The input will consist of multiple test cases.

- The first line of each test case will contain an integer, indicating the value of **k**.
- The following k lines will contain k integers each. The i^{th} integer of the a^{th} line among these k lines will indicate the price of the i^{th} item in section a.
- The last line of the input will be indicated by an integer -1. No test cases will come after this.

5.2 Output

A single line of output for every test case. Each line should have k space-separated integers showing the price of the cheapest k orders in ascending order.

5.3 Constraints

$1 \leq k \leq 800$
$1 \leq price\ of\ one\ item \leq 10^6$

5.4 Example

Input	Output
2 5674 7577 6202 9191 5 13 48 86 76 70 36 95 31 80 56 46 1 93 92 61 66 94 76 15 41 93 33 53 97 89 -1	11876 13779 93 98 113 118 118

6 Submission format

Online Judge Submission

You need to create a single *.c file for each question. From the previous assignments, you should be already comfortable with merging individual files of your solutions to a single $< filename > .c$ file for submission on OJ.