Islamic University of Technology
Department of Computer Science and Engineering

Lab 2: Sorting

CSE 4404: Algorithms Lab
Summer 2023-24

Task A. The Great Bottle Fill-Off

Time Limit: 1 second | Memory Limit: 256 MB

You are given a container filled with a limited amount of water. The container can hold at most x liters of water. You are also given n empty bottles, each with a certain capacity. You want to fill as many bottles as possible using the water in the container, such that each bottle is filled to its full capacity.

Your task is to determine the **maximum number of bottles** you can completely fill using the water from the container.

Input Format

The first line contains a single integer t ($1 \leq t \leq 10^5$) — the number of test cases.

The first line of each test case contains two integers n and x ($1 \leq n \leq 10^4,\ 1 \leq x \leq 10^9$) — the number of bottles and the total capacity of the container in liters.

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 required amount of water to completely fill each bottle.

The sum of n over all test cases does not exceed 10^5.

Output Format

For each test case, print a single integer in a line — the maximum number of bottles you can completely fill.

Examples

Sample Input	Sample Output
3 5 10 2 3 4 5 1 4 7 5 3 4 2 3 5 6 7 8	4 2 0

Task B. Dot Product Minimization

Time Limit: 1 second | Memory Limit: 256 MB

You are given two vectors **a** and **b**, each containing n integers. You are allowed to rearrange the elements of each vector in any order you like. Your goal is to find such rearrangements that **minimize the dot product** of the two vectors.

Formally, given two vectors $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n)$, find permutations π and σ of $\{1, 2, \ldots, n\}$ such that the value

$$\sum_{i=1}^{n} a_{\pi(i)} \cdot b_{\sigma(i)}$$

is minimized.

Input Format

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

For each test case, the first line contains a single integer n ($1 \le n \le 10^5$) — the size of the vectors.

The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ ($|a_i| \le 10^9$) — the elements of the first vector.

The third line contains n space-separated integers $b_1, b_2, \ldots, b_n$ ($|b_i| \le 10^9$) — the elements of the second vector.

The sum of n over all test cases does not exceed 10^5.

Output Format

For each test case, print a single integer — the **minimum possible dot product** after rearranging the vectors.

Examples

Sample Input	Sample Output
4 3 1 3 -5 -2 4 1 4 1 2 3 4 5 6 7 8 5 -1 -2 -3 -4 -5 5 4 3 2 1 5 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000 1000000	-25 60 -55 5000000000000

Task C. Plan It Right

Time Limit: 1 second | *Memory Limit: 256 MB*

You are an event planner hired to organize a series of activities for a group of tourists visiting a city. You are given a list of n activities, each with a start time and an end time. A tourist can participate in an activity only if they are not already engaged in another one at that time.

Your goal is to select a subset of non-overlapping activities such that the tourists can attend the **maximum number of activities**.

Input Format

The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

For each test case, the first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of activities.

The second line contains n space-separated integers $s_1, s_2, \ldots, s_n$ ($1 \leq s_i \leq 10^9$) — the start times of the activities.

The third line contains n space-separated integers $e_1, e_2, \ldots, e_n$ ($1 \leq e_i \leq 10^9$, $s_i \leq e_i$) — the end times of the activities.

The sum of n over all test cases does not exceed 10^5.

Output Format

For each test case, print a single integer — the maximum number of non-overlapping activities that the tourists can attend.

Examples

Sample Input	Sample Output
5 4 0 5 1 3 6 7 2 4 3 1 4 6 5 7 10 6 3 8 2 1 5 3 9 9 5 4 9 6 5 1 2 3 1 4 10 10 5 2 5 3 1 6 4 5 10 7	3 2 2 2 2

Marks Distribution

Task	**Marks**
Task A	40%
Task B	40%
Task C	20%