Bus Queues

Time Limit: 1 s
Memory Limit: 256 MB

Description

The Bus Queues problem involves working with multiple bus queues, where you need to perform certain operations on them. Given a set of N bus queues, you will be asked to populate each queue with integer values indicating the capacity of the bus. Unfortunately, you cannot use all of the buses so you will be given a number K, which represents the number of buses needed. You need to get the buses with max capacity at the front of the bus queues to maximise the profits.

Input

- The first line contains an integer N, representing the number of bus queues.
- For each of the N queues, the input will be provided in the following format:
    - The first line contains an integer M, representing the number of buses to be stored in the current queue.
    - The next M lines contain one integer value each, representing the capacity for each bus to be stored in the current queue.
- The last line contains an integer K, representing the number of buses needed to operate.

Output

- The output should be a single line containing the total sum of the buses capacity that operates

Constraints

- *If There are more than 1 queue that have same max, prioritise the queue with lower index*
- $N < 100$

ONLY ACCEPT SOLUTIONS THAT CONTAINS DYNAMIC ARRAY / LINKED LIST

Example #1

Input
3 3 1 2 3 2 4 5 1 6 3
Output
15

Explanation #1

GET THE BUSES WITH MAX VALUE FROM THE FRONT QUEUES

Queue #1: 1 - 2 - 3
Queue #2: 4 - 5
Queue #3: 6
Bus Needed: 3

1st Bus: Get **6** From Queue #3
2nd Bus: Get **4** From Queue #2
3rd Bus: Get **5** From Queue #2
Total: **15**

Example #2

Input
4 2 40 50 3 10 80 90 1 60 2 70 20 4
Output
220

Explanation #2

GET THE BUSES WITH MAX VALUE FROM THE FRONT QUEUES

Queue #1: 40 - 50
Queue #2: 10 - 80 - 90
Queue #3: 60
Queue #4: 70 - 20
Bus Needed: 4

1st Bus: Get **70** From Queue #4
2nd Bus: Get **60** From Queue #3
3rd Bus: Get **40** From Queue #1
4th Bus: Get **50** From Queue #1
Total: **220**