Problem A. Another Card Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Ishu and Rohu are back with their exciting card game using a deck of N cards. Each card has a non-negative value assigned to it.

The game is played as follows: Ishu and Rohu take turns drawing cards from the deck. In each turn, Rohu draws the card with the smallest value, while Ishu draws the card with the largest value. After each draw, a new card is placed in the deck with a value equal to the absolute difference between the cards drawn by Ishu and Rohu. This process continues until there are fewer than 2 cards left in the deck.

After each turn, Ishu and Rohu want to know the score, which is the sum of all the card values obtained up to that point.

You need to write a program to help Ishu and Rohu determine the score after each turn.

Input

The first line of input contains a single integer, N ($1 \leq N \leq 10^5$) denoting the number of cards.

The second line of input contains N space-separated non-negative integers, representing the initial values assigned to the N cards. The values ranges from 1 to 10^9.

Output

Print $n - 1$ lines, each containing a single integer. The i-th line represents the score after the i-th turn.

Examples

standard input	standard output
6 8 9 2 10 5 4	34 26 16 10 0
8 1 2 3 4 5 6 7 8	34 30 24 16 12 6 2

Note

Explanation for Testcase-1

1. Initial cards value: $[8, 9, 2, 10, 5, 4]$.

   Score: $8 + 9 + 2 + 10 + 5 + 4 = 38$ (sum of all elements).

2. After the first turn:

   - Remove the largest element 10 and the smallest element 2.
   - Calculate the difference: $10 - 2 = 8$.
   - Add the difference back into the card values: $[8, 9, 5, 4, 8]$.

- Sum of the card values: $8 + 9 + 5 + 4 + 8 = 34$.
- Updated card values: $[8, 9, 5, 4, 8]$.
  Score: 34.

3. After the second operation:
   - Remove the largest element 9 and the smallest element 4.
   - Calculate the difference: $9 - 4 = 5$.
   - Add the difference back into the card values: $[8, 5, 8, 5]$.
   - Sum of the card values: $8 + 5 + 8 + 5 = 26$.
   - Updated card values: $[8, 5, 8, 5]$.
     Score: 26.
4. After the third turn:
   - Remove the largest element 8 and the smallest element 5.
   - Calculate the difference: $8 - 5 = 3$.
   - Add the difference back into the card values: $[8, 5, 3]$.
   - Sum of the card values: $8 + 5 + 3 = 16$.
   - Updated card values: $[8, 5, 3]$.
     Score: 16.
5. After the fourth turn:
   - Remove the largest element 8 and the smallest element 3.
   - Calculate the difference: $8 - 3 = 5$.
   - Add the difference back into the card values: $[5, 5]$.
   - Sum of the card values: $5 + 5 = 10$.
   - Updated card values: $[5, 5]$.
     Score: 10.
6. After the fifth turn:
   - Remove the largest element 5 and the smallest element 5.
   - Calculate the difference: $5 - 5 = 0$.
   - Add the difference back into the card values: $[0]$.
   - Sum of the card values: 0.
   - Updated card values: $[0]$.
     Score: 0.

   Since only one card is left so game will stop here.

Problem B. Farmer and the Goats

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

NOTE - you need to implement a heap from scratch. Also note that you are not allowed to use libraries other than stdlib.h and stdio.h your submission will not be graded if implementation of heap is not found.

Sandheap is a farmer who owns a herd of N goats. He goes everyday to a nearby rental farm to feed hay and grass to the goats. Since every farm has different capacity to accommodate a number of goats, therefore let's say the farm they go to has a capacity to accommodate K goats at a time. Each goat takes some specific time to eat, say i^{th} goat takes T_i time. Once a goat is done eating, immediately it moves out and another goat comes in. So let's say that the total time taken till the last goat has eaten is T. The farmer has to feed the goats in at most T_u time since the farm is booked on a rental basis. Therefore he must go to a farm with enough size to accommodate the goats so that he gets done well within time. Your task is to determine the infimum value of K so that he can rent a farm with the optimal constraints.

Constraints:

- $1 \leq N \leq 10^4$
- $1 \leq T_U \leq 10^6$
- $1 \leq T_i \leq 10^5$

Input

The first line contains 2 spaced integers N and T_u.
Following N lines contains N integers denoting T_i i.e. the time taken by each goat to eat the hay and grass.

Output

Print the infimum value of K such that all goats are fed in $\leq T_u$ time.

Example

standard input	standard output
5 8 4 7 8 6 4	4

Problem C. Kone Visits Art Galleries

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Kone has finally graduated and being passionate about art, wants to celebrate by visiting art galleries. Kone finds there are n art galleries in his area and wants to visit as many as he can within k days. Now, the i^{th} art gallery has w^i paintings and Kone wants to see the maximum number of paintings in each art gallery. Therefore, if Kone sees a^i paintings at the i^{th} art gallery, he wants $w^i - a^i$ to be minimum for all art galleries.

To come up with an efficient strategy, he seeks the wisdom of his pals Lawrence and TarS. The group can visit atmost 1 art gallery per day. They propose an algorithm - if the paintings left to be seen for a given art gallery are w, then Kone only sees $ceil(\frac{w}{2})$ paintings and hope to revisit the art gallery later.

Now, TarS knows that Lawrence dreads paintings and if the group sees a^i paintings for the i^{th} art gallery, Lawrence will be bored by $b^i * a^i$ units. But Kone would still prefer seeing maximum number of paintings, however, if there are multiple optimal ways to go about it, he would be considerate of Lawrence's feelings and take the approach which bores Lawrence the least.

Now, it is up to you to help Kone, Lawrence and TarS. Formally, you have to maximise the total number of paintings they can see and if there are multiple answers, choose the one which minimises Lawrence's boredom.

Input

- the first line contains the number of test cases ($1 \leq t \leq 5$)
- the first line of each test case contains two integers n and k indicating the number of art galleries and days available ($1 \leq n, k \leq 10^5$)
- the second line of each test case contains n space separated integers indicating the number of paintings in each art gallery ($1 \leq w^i \leq 10^6$)
- the third line of each test case contains n space separated integers indicating Lawrence's boredom for completing 1 painting in the i^{th} art gallery ($1 \leq b^i \leq 10^6$)

Output

For each test case, print two space separated integers - the total number of paintings left to be seen and the total boredom acquired by Lawrence.

Example

standard input	standard output
1 4 4 1 2 3 4 5 3 1 1	4 6

Note

In the above sample case: On day 1, they see ceil(3/2) = 2 paintings of art gallery 3. On day 2, they see ceil(4/2) = 2 paintings of art gallery 4. On day 3, they see ceil(2/2) = 1 paintings of art gallery 4. On day 4, they see ceil(1/2) = 1 paintings of art gallery 3.

Total paintings seen are 2+2+1+1 = 6 and paintings left are 1+2+3+4-6 = 4. The corresponding boredom value is 2*1 + 2*1 + 1*1 + 1*1 = 6 units.

Problem D. Kone Meets Aliens

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Kone recently joined forces with Melon Tusk and went to the first ever mission on Mars, and much to his delight he made friends with aliens over there. Now Kone has to rank all these aliens based on their hostility and humour. Kone collects this data for n aliens and sends it over to Earth for his friend Lawrence to analyze. To be specific, Kone sends q queries to Lawrence, where each query contains a single integer q^i and he's supposed to report the q^i_{th} alien in the increasing order of the sum of hostility + humour values. Help Lawrence and Kone with this task and play a role in strengthening Earth-Mars ties.

Update: the sum to be considered corresponds to all possible combinations of hostility and humour.

Input

- the first line contains two space separated integers - the number of aliens n and the number of queries q ($1 \leq n \leq 20000$ and $1 \leq q \leq 500$)
- the second line contains n elements representing the hostility value of each alien ($1 \leq a^i \leq 10^{18}$)
- the third line contains n elements representing the humour value of each alien ($1 \leq b^i \leq 10^{18}$)
- then q lines follow where each line contains a single integer q^i ($1 \leq q^i \leq 20000$)

Output

For each query, report the q^i_{th} value of hostility + humour sum in increasing order

Example

standard input	standard output
3 2 1 2 3 6 7 8 3 4	8 9

Note

The 9 elements in the set sum are:
1+6=7, 1+7=8,
2+6=8,
1+8=9,
2+7=9,
3+6=9,
2+8=10,
3+7=10,
3+8=11

Problem E. AVL Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are tasked with implementing an AVL tree, a self-balancing binary search tree. The tree supports two types of operations: `insert x` and `subtree x`.

- For the `insert x` **operation, you need to insert the element x into the AVL tree. After the insertion, you should print two space-separated integers representing the total number of left rotations (L) and right rotations (R) performed during the insertion process, respectively.**
- **The `subtree x` operation requires you to print the sum of all elements in the subtree rooted at node x, including the value of x itself if x exists otherwise print 'NA'.**

Constraints:

- The number of queries, denoted by Q, will be between 1 and 10^5, inclusive. ($1 \leq Q \leq 10^5$)
- Each query will be either of the two types: `insert x` or `subtree x`.
- For the `insert x` operation, the value of x will be an integer between -10^9 and 10^9, inclusive. ($-10^9 \leq x \leq 10^9$)
- All input and output values will fit in a 64-bit signed integer.

Input

The first line of the input contains an integer Q, representing the number of queries.

The next Q lines contain the queries. Each query is in one of the following formats:

- **insert x: representing the insertion of the element x into the AVL tree.**
- **subtree x: representing the query to print the sum of the subtree rooted at node x.**

It is guaranteed that the input format will follow the constraints mentioned earlier.

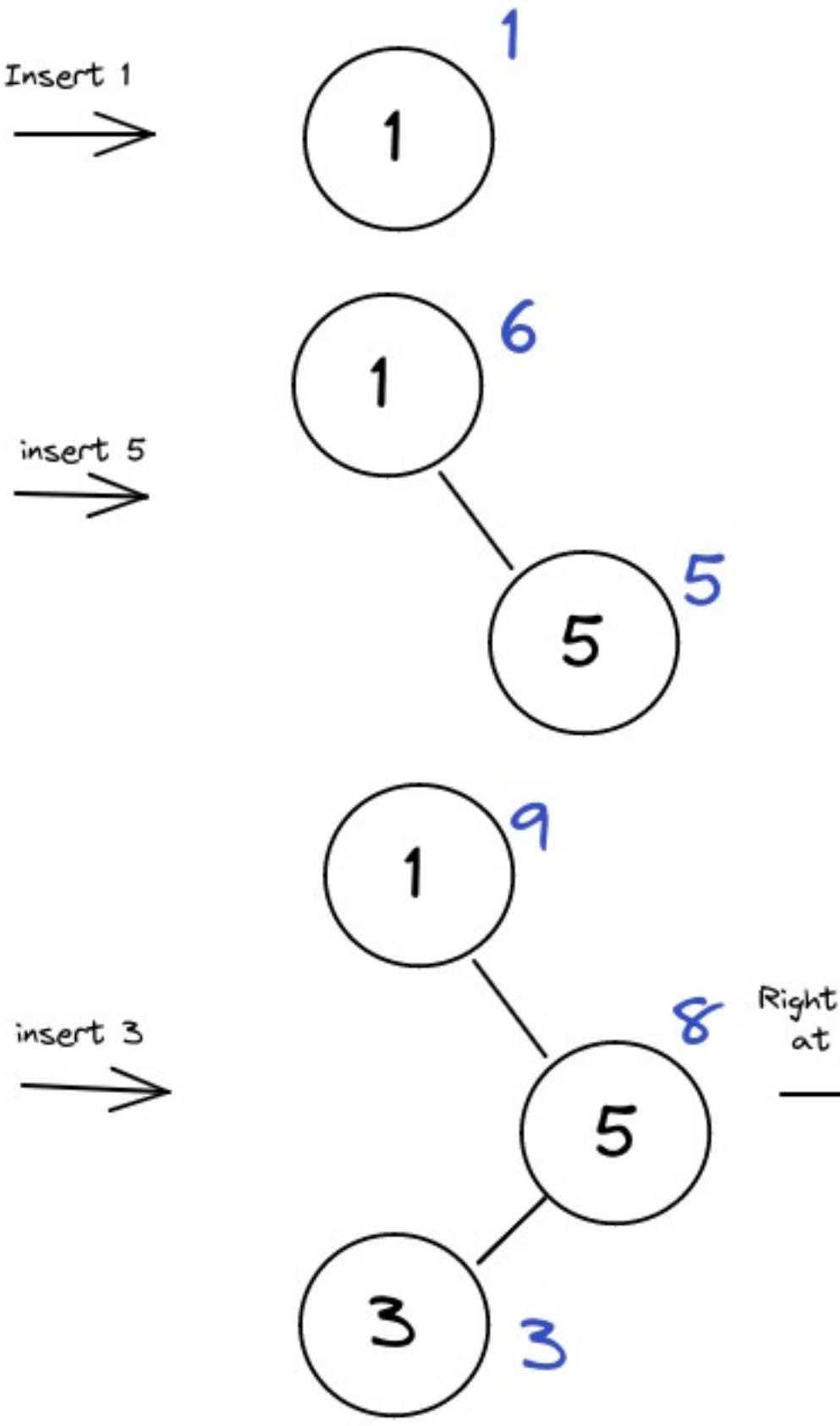
Output

print the answer of each query in above mentioned format.

Examples

standard input	standard output
9 insert 1 subtree 1 insert 5 subtree 1 insert 3 subtree 1 subtree 3 subtree 5 subtree 10	0 0 1 0 0 6 1 1 1 9 5 NA
8 insert 10 insert 13 subtree 10 insert 9 insert 8 subtree 9 insert 7 subtree 9	0 0 0 0 23 0 0 0 0 17 0 1 9

Note



Explanation of Testcase-1