Lab – 09
23/03/2025

Tree ADT – Binary Search Tree

Note:

1. Use only Visual Studio code type your program and run your code.
2. Always follow industry coding best practices.

A. Utilize C++ STL to solve the following (K5),

Nene invented a new game based on an increasing sequence of integers $a_1, a_2, \ldots, a_k$.

In this game, initially n players are lined up in a row. In each of the rounds of this game, the following happens:

- Nene finds the a_1-th, a_2-th, ..., a_k-th players in a row. They are kicked out of the game simultaneously. If the i-th player in a row should be kicked out, but there are fewer than i players in a row, they are skipped.

Once no one is kicked out of the game in some round, all the players that are still in the game are declared as winners.

For example, consider the game with $a = [3, 5]$ and $n = 5$ players. Let the players be named player `A`, player `B`, ..., player `E` in the order they are lined up initially. Then,

- Before the first round, players are lined up as `ABCDE`. Nene finds the 3-rd and the 5-th players in a row. These are players `C` and `E`. They are kicked out in the first round.
- Now players are lined up as `ABD`. Nene finds the 3-rd and the 5-th players in a row. The 3-rd player is player `D` and there is no 5-th player in a row. Thus, only player `D` is kicked out in the second round.
- In the third round, no one is kicked out of the game, so the game ends after this round.
- Players `A` and `B` are declared as the winners.

Nene has not yet decided how many people would join the game initially. Nene gave you q integers $n_1, n_2, \ldots, n_q$ and you should answer the following question for each $1 \leq i \leq q$ **independently**:

- How many people would be declared as winners if there are n_i players in the game initially?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 250$). The description of test cases follows.

The first line case contains two integers k and q ($1 \le k, q \le 100$) — the length of the sequence a and the number of values n_i you should solve this problem for.

The second line contains k integers $a_1, a_2, \ldots, a_k$ ($1 \le a_1 < a_2 < \ldots < a_k \le 100$) — the sequence a.

The third line contains q integers $n_1, n_2, \ldots, n_q$ ($1 \le n_i \le 100$).

Output

For each test case, output q integers: the i-th ($1 \le i \le q$) of them should be the number of players declared as winners if initially n_i players join the game.

Example

input

Copy

```
6
2 1
3 5
5
5 3
2 4 6 7 9
1 3 5
5 4
3 4 5 6 7
1 2 3 4
2 3
69 96
1 10 100
1 1
100
50
3 3
10 20 30
1 10 100
```

output

Copy

```
2
1 1 1
1 2 2 2
1 10 68
50
1 9 9
```

Note

The first test case was explained in the statement.

In the second test case, when $n = 1$, the only player stays in the game in the first round. After that, the game ends and the only player is declared as a winner.

B. Utilize C++ STL to solve the following (K5),

There are n participants in a competition, participant i having a strength of s_i.

Every participant wonders how much of an advantage they have over the other best participant. In other words, each participant i wants to know the difference between s_i and s_j, where j is the strongest participant in the competition, not counting i (a difference can be negative).

So, they ask you for your help! For each i ($1 \leq i \leq n$) output the difference between s_i and the maximum strength of any participant other than participant i.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 1000$) — the number of test cases. The descriptions of the test cases follow.

The first line of each test case contains an integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of the array.

The following line contains n space-separated positive integers s_1, s_2, ..., s_n ($1 \leq s_i \leq 10^9$) — the strengths of the participants.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output n space-separated integers. For each i ($1 \leq i \leq n$) output the difference between s_i and the maximum strength of any other participant.

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \leq t \leq 1000$) — the number of test cases. The descriptions of the test cases follow.

The first line of each test case contains an integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of the array.

The following line contains n space-separated positive integers s_1, s_2, ..., s_n ($1 \leq s_i \leq 10^9$) — the strengths of the participants.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output n space-separated integers. For each i ($1 \leq i \leq n$) output the difference between s_i and the maximum strength of any other participant.

Example

input Copy

```
5
4
4 7 3 5
2
1 2
5
1 2 3 4 5
3
4 9 4
4
4 4 4 4
```

output Copy

```
-3 2 -4 -2
-1 1
-4 -3 -2 -1 1
-5 5 -5
0 0 0 0
```

Note

For the first test case:

- The first participant has a strength of 4 and the largest strength of a participant different from the first one is 7, so the answer for the first participant is $4 - 7 = -3$.
- The second participant has a strength of 7 and the largest strength of a participant different from the second one is 5, so the answer for the second participant is $7 - 5 = 2$.
- The third participant has a strength of 3 and the largest strength of a participant different from the third one is 7, so the answer for the third participant is $3 - 7 = -4$.
- The fourth participant has a strength of 5 and the largest strength of a participant different from the fourth one is 7, so the answer for the fourth participant is $5 - 7 = -2$.

C. Write a separate C++ menu-driven program to implement Tree ADT using a binary search tree. Maintain proper boundary conditions and follow good coding practices. The Tree ADT has the following operations,

1. Insert
2. Preorder
3. Inorder
4. Postorder
5. Search
6. Exit

What is the time complexity of each of the operations? **(K4)**

D. Add a "construct expression tree" method to the binary tree data structure from the previous lab code—import stack from the standard template library (STL) to construct the expression tree. Import the Tree ADT program into another program that gets a valid postfix expression, constructs, and prints the expression tree. It consists of the following operations.

1. Postfix Expression
2. Construct Expression Tree
3. Preorder
4. Inorder
5. Postorder
6. Exit

What is the time complexity of each of the operations? **(K4)**