

**Lab – 10**  
**31/03/2025**

**Priority Queue ADT – Heap**

**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 if needed. Write a C++ program to solve the following,

Polycarp was presented with some sequence of integers  $a$  of length  $n$  ( $1 \leq a_i \leq n$ ). A sequence can make Polycarp happy only if it consists of **different** numbers (i.e. distinct numbers).

In order to make his sequence like this, Polycarp is going to make some (possibly zero) number of moves.

In one move, he can:

- remove the first (leftmost) element of the sequence.

For example, in one move, the sequence  $[3, 1, 4, 3]$  will produce the sequence  $[1, 4, 3]$ , which consists of different numbers.

Determine the minimum number of moves he needs to make so that in the remaining sequence all elements are different. In other words, find the length of the smallest prefix of the given sequence  $a$ , after removing which all values in the sequence will be unique.

**Input**

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

Each test case consists of two lines.

The first line contains an integer  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) — the length of the given sequence  $a$ .

The second line contains  $n$  integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — elements of the given sequence  $a$ .

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

**Output**

For each test case print your answer on a separate line — the minimum number of elements that must be removed from the beginning of the sequence so that all remaining elements are different.

### Example

input	Copy
5 4 3 1 4 3 5 1 1 1 1 1 1 1 6 6 5 4 3 2 1 7 1 2 1 7 1 2 1	
output	Copy
1 4 0 0 5	

### Note

The following are the sequences that will remain after the removal of prefixes:

- [1, 4, 3];
- [1];
- [1];
- [6, 5, 4, 3, 2, 1];
- [2, 1].

It is easy to see that all the remaining sequences contain only distinct elements. In each test case, the shortest matching prefix was removed.

## B. Utilize C++ STL if needed. Write a C++ program to solve the following,

Three guys play a game: first, each person writes down  $n$  distinct words of length 3. Then, they total up the number of points as follows:

- if a word was written by one person — that person gets 3 points,
- if a word was written by two people — each of the two gets 1 point,
- if a word was written by all — nobody gets any points.

In the end, how many points does each player have?

### Input

The input consists of multiple test cases. The first line contains an integer  $t$  ( $1 \leq t \leq 100$ ) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer  $n$  ( $1 \leq n \leq 1000$ ) — the number of words written by each person.

The following three lines each contain  $n$  **distinct** strings — the words written by each person. Each string consists of 3 lowercase English characters.

### Output

For each test case, output three space-separated integers — the number of points each of the three guys earned. You should output the answers in the same order as the input; the  $i$ -th integer should be the number of points earned by the  $i$ -th guy.

### Example

input	Copy
<pre>3 1 abc def abc 3 orz for qaq qaq orz for cod for ces 5 iat roc hem ica lly bac ter iol ogi sts bac roc lly iol iat</pre>	
output	Copy
<pre>1 3 1 2 2 6 9 11 5</pre>	

### Note

In the first test case:

- The word **abc** was written by the first and third guys — they each get **1** point.
- The word **def** was written by the second guy only — he gets **3** points.

C. Write a separate C++ menu-driven program to implement Priority Queue ADT using a max heap. Maintain proper boundary conditions and follow good coding practices. The Priority Queue ADT has the following operations,

1. Insert
2. Delete
3. Display
4. Search
5. Sort (Heap Sort)
6. Exit

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