CSE 4304-Data Structures Lab. Winter 2022-23
Batch: CSE 21
Date: September 4, 2023,
Target Group: All
Topic: Strings, sorting, binary search, Stack, Queue, Deque

Instructions:

- Regardless you finish the tasks in the lab, you have to submit the solutions in the Google Classroom. In case I forget to upload the tasks there, CR should contact me. The deadline will always be at 11.59 PM of the day in which the lab has taken place.
- Task naming format: fullID_T01L02_2A.c/cpp
- If you find any issues in the problem description/test cases, comment in the Google Classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Use appropriate comments in your code. This will help you to easily recall the solution in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use <bits/stdc++.h> library.
- Modified sections will be marked with BLUE color.

Group	Tasks
2A	1 3
2B	1 2
1A	
1B	

Task 1 – Unethical Queue

Problem Statement

It is considered a bad manner to cut in front of a line of people because it is unfair to those at the back. However, we see such unethical behavior very regularly in our daily lives. Let's call such a queue an ***Unethical Queue***.

In an unethical queue, each element belongs to a friend circle. If an element (person) enters the queue, he first searches the queue from head to tail to check if some of his friends (elements of the same friend circle) are already in the queue. If yes, he enters the queue right behind them. If not, he enters the queue at the tail and becomes the new last element. Dequeuing is done like in normal queues– elements are processed from head to tail in the order they appear in the unethical queue.

Your task is to write a program that simulates such an unethical queue.

Input

The input will contain one or more test cases. Each test case begins with the number of friend circles t. Then t friend circle descriptions follow, each one consisting of the number of elements belonging to the group and the elements themselves. A friend group may contain many people/elements.

Finally, a list of commands follows. There are three different kinds of commands:

1. **ENQUEUE** x - enter person x into the unethical queue
2. **DEQUEUE** - process the first element and remove it from the queue
3. **STOP** - end of test case

The input will be terminated by a value of 0 for t.
Note: The implementation of **the unethical queue should be efficient** – both enqueuing and dequeuing of an element should only take **constant time**.

Output

For each test case, first print a line saying "Scenario #k", where k is the number of the test case. Then, for each DEQUEUE command, print the element which is dequeued on a single line. Print a blank line after each test case, even after the last one.

Sample Test Case(s)

Input

```
2
3 101 102 103
3 201 202 203
ENQUEUE 101
ENQUEUE 201
```

```
ENQUEUE 102
ENQUEUE 202
ENQUEUE 103
ENQUEUE 203
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
STOP
2
5 259001 259002 259003 259004 259005
6 260001 260002 260003 260004 260005 260006
ENQUEUE 259001
ENQUEUE 260001
ENQUEUE 259002
ENQUEUE 259003
ENQUEUE 259004
ENQUEUE 259005
DEQUEUE
DEQUEUE
ENQUEUE 260002
ENQUEUE 260003
DEQUEUE
DEQUEUE
DEQUEUE
DEQUEUE
STOP
0
```

Output

```
Scenario #1
101
102
103
201
202
203

Scenario #2
259001
259002
259003
259004
259005
260001
```

Task 2 – Maximum Effort

Problem Statement

You are given an array A and an integer k. Your task is to find the maximum value within each contiguous subarray of size k.

Input

The first line of the input is the number of test cases t. Then, t lines follow. The first line of each test case consists of n and k, the size of the array A and the subarray size respectively. The second line consists of the array elements $A_1, A_2, \ldots A_n$.

Output

For each test case, find the maximum for each and every contiguous subarray of size k.

Sample Test Case(s)

Input

```
3
9 3
1 2 3 1 4 5 2 3 6
10 4
8 5 10 7 9 4 15 12 90 13
8 3
1 3 -1 -3 5 3 6 7
```

Output

```
3 3 4 5 5 5 6
10 10 10 15 15 90 90
3 3 5 5 6 7
```

Note

Your solution must be O(n).

Task 3 – Largest Rectangle in a Bar Graph

Problem Statement

A bar graph is a geometric shape formed by a series of rectangles placed along a shared baseline.
These rectangles have uniform widths but can vary in their individual heights. For instance, consider the illustration on the left side of Figure-1, which depicts a histogram comprising rectangles with heights of 2, 1, 4, 5, 1, 3, and 3 units, respectively, where each rectangle's width is equivalent to 1 unit.

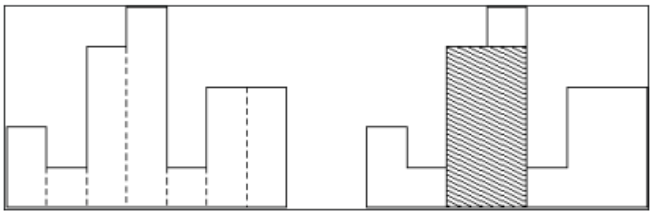

Figure 1: An example bar chart with its largest rectangle.

Calculate the area of the largest rectangle in a bar graph that is aligned at the common base line, too. The illustration on the right side of Figure-1 shows the largest aligned rectangle for the depicted bar graph.

Input

The input contains several test cases. Each test case describes a bar graph and starts with an integer n, denoting the number of rectangles it is composed of. Then follow n integers h_1, ..., h_n. These numbers denote the heights of the rectangles of the bar graph in left-to-right order. The width of each rectangle is 1. **A zero follows the input for the last test case.**

Output

For each test case output on a single line the area of the largest rectangle in the specified bar graph. Remember that this rectangle must be aligned at the common base line.

Sample Test Case(s)

Input

```
7 2 1 4 5 1 3 3
4 1000 1000 1000 1000
3 2 1 2
8 2 1 2 0 3 2 2 3
1 0
0
```

Output

```
8
4000
3
8
0
```