Homework #2

RELEASEDATE: 2025/11/03

DUEDATE:2025/11/24, 23:59 on iLearning 3.0

1. *You need to submit your code to the designated place on iLearning 3.0.*
2. *As for coding, C and C++ are allowed.*
3. *Discussions on course materials and homework solutions are encouraged. But you should write the final solutions alone and understand them fully. Books, notes, and Internet resources can be consulted, but not copied from.*
4. *For each question, there will be a testcase (e.g. the testcase for question 1 is testcase1.txt) Please note that the testcase you received may not necessarily to be the same as the testcase used for grading. Please consider edge cases.*
5. *Let your program read the corresponding testcase and print the result to a corresponding text file (e.g. the output for question 2 should be output2.txt).*
6. *For each question, we have provided a corresponding answer (e.g assume you're solving problem 3, the corresponding answer is ans3.txt), and your output should be same as answer.*
7. *For each question, your program file should be named DS{problem number}_{student ID}.c. (e.g. assume your student ID is 71140561387 and you're solving question 3, your program file should be DS3_7114056138.c)*
8. *Each question is scored independently, but partial credit is not awarded; full credit is given only for complete correctness.*
9. *If you have any questions, please feel free to email the TA at any time.*

Teaching Assistant:

- Name: 黃意丞
- Email: g114056138@mail.nchu.edu.tw
- Lab: WCCCLab (S901)

1 Check the Order (10 pt.)

Given a number **N**, determine whether a given permutation of the numbers **1, 2, ..., N** is a valid **stack permutation**.

- This question is the same as the previous assignment.
- You must use the following structure, or else you will receive a score of 0.
- You shouldn't modify the fields in Node or Stack. However, you can modify the values if you want.

```
typedef struct Node {
    // you can add data type you want
    struct Node *next;
} Node;

typedef struct {
    // you can add data type you want
    Node *top;
} Stack;
```

Input

The first line of input gives the number of test cases, T (1≤T≤50). Then T test cases follow each described in the following way. The first line of each test case contains the integer N described above. For each of the next lines of the test case there is a permutation of 1, 2, …, N.

Output

The output contains the lines corresponding to the lines with permutations in the input. A line of the output contains 'Yes' if it is possible to ship out

goods in the order required on the corresponding line of the input. Otherwise, it contains ‘No’.

Sample Input

2

5

1 2 3 4 5

5

5 4 1 2 3

Sample Output

Yes

No

2 Discard the Number (25 pt.)

Given a sequence $\{x_1, x_2, \ldots, x_m\}$, where all numbers in the sequence are positive integers. Perform the following operation on this sequence: start by looking at x_1, then x_2, and continue in this manner. After reaching x_m, start again from x_1, repeating this process cyclically. Each time you look at a number, say x_i, remember it and compare it with the last remembered number x_j. If the two numbers satisfy $\gcd(x_i, x_j) = 1$, discard x_i from the sequence and forget it. Then, restart the process from x_{i+1}. This means that after discarding a number, you cannot immediately discard the next number in the subsequent step.

Your task is to determine which numbers will be discarded and output them in the order they are discarded.

- You need to represent the elements as a doubly circular linked list using the following structure, or else you will receive a score of 0.
- You shouldn't modify the fields in Node or LinkedList. However, you can modify the values if you want.
- Traversing the number and constructing it with an array is not allowed.

```
struct Node {
    int order, num;
    struct Node *next, *prev;
};

struct LinkedList {
    struct Node *head;
};
```

Input

The first line of input gives the number of testcases, T($1 \le T \le 50$). Then T test cases follow each described in the following way. The first line contains a single integer m ($1 \le m \le 5 \times 10^4$) indicates the number of elements. The second line contains the integers $x_1, x_2, ..., x_m$ ($1 \le x_i \le 10^9$) separated by spaces, which indicate the number on elements.

Output

For each input produce one line of output. First, print a single integer k, which indicates the number of discarded elements. After that print k distinct integer: discarded elements in the order of their deletion.

Sample Input

```
5
5
5 9 2 10 15
6
1 2 4 2 4 2
2
1 2
1
1
1
2
```

Sample Output

```
2 2 3
2 2 1
2 2 1
1 1
0
```

3 Determine Base (30 pt.)

Blackbeard, a ruthless and notorious pirate, carries out raids across the seas in pursuit of his own ambitions. As his trusted confidant, you are responsible for his calculations and strategies.

One day, he acquires a nautical chart of an archipelago, and he decides to make these islands his next target of conquest. He needs to choose one island from the archipelago as his base of operations, and it is your task to determine which island is the most suitable, based on the information from the chart.

First, you assign a number to each island to establish how many islands exist on the map, with numbering starting from 1. According to the chart, some routes between islands are too dangerous for sailing, so only the marked safe routes can be used to travel. You may freely move back and forth between islands connected by these safe routes. You also discover that the sailing time between any two connected islands is the same.

Therefore, you decide to classify the islands into two categories: strategic hubs and ordinary islands. A strategic hub is defined as an island that appears on more than one route, and there may be multiple such hubs within the archipelago. From these hubs, you must select one as Blackbeard's base.

The criterion for choosing the base is as follows:

- The average sailing time from the base to all other strategic hubs must be minimized.

• If you use queue in this question, you must use the following structure, or else you will receive a score of 0.

• You can add data type in Node or Queue.

```c
typedef struct Node {
    // you can add data type you want
    struct Node *next, *pre;
} Node;

typedef struct Queue {
    // you can add data type you want
    Node *front, *rear;
} Queue;
```

Input

The first line indicates the number of test cases. For each test case, the first line contains two integers: N and S. N is the total number of islands, and S is the number of routes. Islands are numbered from 1 to N; N is not greater than 10000; and S is not greater than 100. Next, we have S lines, one for each. These lines consist of a list of islands, separated by blank spaces, ending with a '0'. There will be between 1 and 100 hubs, inclusive. There is always a path between any pair of islands.

Output

For each test case, you have to output the number of the resulting hub. If there are more than one hub with the minimum distance, then you have to output the one with the smallest number.

Sample Input

4

13 3

1 2 3 4 5 6 7 0

8 9 4 10 13 0

11 2 12 9 6 7 0

6 2

2 5 3 6 1 4 0

4 1 6 3 5 2 0

5 2

1 2 3 4 5 0

3 5 1 4 2 0

7 2

3 5 1 2 4 7 6 0

3 6 1 0

Sample Output

9

3

4

6

4 The Beast Battle (35 pt.)

In the arena, there are n beast tamers, each numbered from 1 to n. Initially, every tamer owns a beast with a power value denoted as s_1, s_2, …, s_n.

Suppose there are two tamers, numbered i and j. The beast of tamer i can defeat the beast of tamer j in either of the following cases:

- $s_i > s_j$
- $s_i = s_j$ and $i > j$

Next, this group of tamers lets their beasts engage in several rounds of battles. In each round, the tamer whose beast has the highest power can choose one of the following two actions:

- Start a battle – The strongest beast defeats the weakest beast, and its power is reduced by the power value of the defeated beast. The defeated beast then leaves the competition, and the next round begins.
- Stop battling – If the tamer chooses not to start a battle, the competition ends immediately.

Each tamer wants their beast to defeat as many other beasts as possible while avoiding being defeated. Assume that all tamers are perfectly rational and make optimal decisions. Please determine how many tamers will still have their beasts when all the battles are over.

• If you use queue in this question, you must use the following structure, or else you will receive a score of 0.

• You can add data type in Node or Queue.

```c
typedef struct Node {
    // you can add data type you want
    struct Node *next, *pre;
} Node;

typedef struct Queue {
    // you can add data type you want
    Node *front, *rear;
} Queue;
```

Input

The first line of input gives the number of testcases, T($1 \le T \le 50$). Then T test cases follow each described in the following way. The first line contains a single integer n($1 \le n \le 5 \times 10^4$) indicates the number of tamers. The second line contains the integers $s_1, s_2, \ldots, s_n$ ($0 \le s_i \le 10^9$) separated by spaces, which indicate the power of beasts. In each testcase, s_i is guaranteed to be arranged in non-decreasing order.

Output

For each input produce one line of output. This line contains an integer which indicates the number beasts will be left when the battles are over.

Sample Input

2

5

13 31 33 39 42

5

7 10 24 48 50

Sample Output

5

3

Submission File:

Please submit the homework to iLearning 3.0 before the deadline. You need to upload your coding part as a single ZIP compressed file to iLearning 3.0 before the deadline. The zip file should be named DS_2_{student ID}.zip (e.g assume your student ID is 7114056138, your zip file should be DS_2_7114056138.zip). The zip file should contain no directories and only the following items:

- all the source code
- an optional README, anything you want the TAs to read before grading your code