

Homework #2

RELEASE DATE: 2025/04/08

DUE DATE: 2025/04/29, 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++** (without lib such as STL) is 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, your program file should be named DS{problem number}_{student ID}.c. (e.g. assume your student ID is 7112056067 and you're solving question 3, your program file should be DS3_7112056067.c)*
7. *Each question is scored independently, but partial credit is not awarded; full credit is given only for complete correctness.*
8. *If you have any questions, please feel free to email the TA at any time.*

Teaching Assistant:

- Name: 王璽銘
- Email: g113056028@mail.nchu.edu.tw
- Lab: WCCCLab (S901)

1 Problem 1

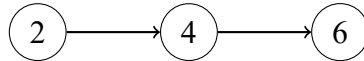
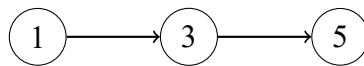
Two departments have each compiled a list of potential recruits. Each list is organized as a singly linked list in **ascending order** of a candidate's score.

Your task is to **merge the two lists** into a single sorted linked list and print out the result.

- You need to represent the linked list using the following structure, or else you will receive a score of 0.
- You should not create new nodes. You must merge the two lists by reusing the original nodes.
- Traversing the list and using arrays or other data structures is not allowed.

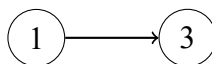
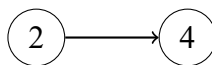
```
1 struct ListNode {
2     int val;
3     struct ListNode *next;
4 };
```

Example 1:



Output: 1 2 3 4 5 6

Example 2:



Output: 1 2 3 4

Input

The first line of input gives the number of test cases, T ($1 \leq T \leq 50$). For each test case:

- First line: a single integer n ($0 \leq n \leq 10^4$) —the length of the first list.
- Second line: n integers in increasing order.
- Third line: a single integer m ($0 \leq m \leq 10^4$) —the length of the second list.

- Fourth line: m integers in increasing order.

Output

For each test case, print the merged list in a single line of space-separated integers.

Sample Input

```
2
3
1 3 5
3
2 4 6
2
2 4
2
1 3
```

Sample Output

```
1 2 3 4 5 6
1 2 3 4
```

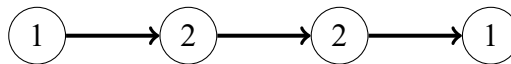
2 Problem 2

Given a singly linked list, print 1 if it is a palindrome or 0 otherwise.

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

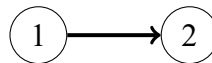
```
1 struct ListNode {  
2     int val;  
3     struct ListNode *next;  
4 };
```

Example 1:



Output: 1

Example 2:



Output: 0

Input

The first line of input gives the number of test cases, T ($1 \leq T \leq 50$). Then T test cases follow each described in the following way. The first line contains a single integer n ($1 \leq n \leq 5 \times 10^3$) indicates the number of nodes. The second line contains the integers a_1, a_2, \dots, a_n separated by spaces, which indicate the number on the node.

Output

For each input produce one line of output. Print 1 if it is a palindrome or 0 otherwise.

Sample Input

```
2
4
1 2 2 1
2
1 2
```

Sample Output

```
1
0
```

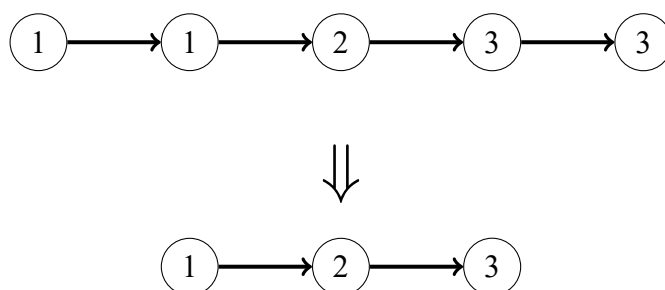
3 Problem 3

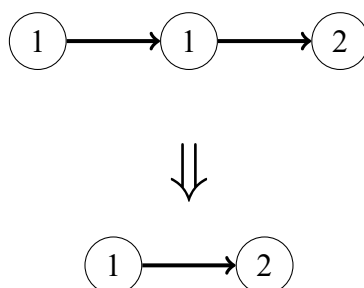
Given a sorted linked list, delete all duplicates such that each element appears only once. Return the linked list sorted as well.

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

```
1 struct ListNode {
2     int val;
3     struct ListNode *next;
4 };
```

Example 1:



Example 2:**Input**

The first line of input gives the number of test cases, T ($1 \leq T \leq 50$). Then T test cases follow each described in the following way. The first line contains a single integer n ($1 \leq n \leq 5 \times 10^3$) indicates the number of nodes. The second line contains the integers a_1, a_2, \dots, a_n separated by spaces, which indicate the number on the node.

Output

For each input produce one line of output. Print the linked list after removing duplicates in order.

Sample Input

```
2
5
1 2 2 3 3
3
1 1 2
```

Sample Output

```
1 2 3
1 2
```

4 Problem 4

The Card Master is playing with patterns of numbers on a circular table of cards. He places m cards in a circle, each marked with a positive integer. Beginning with the first card, he examines cards one by one in clockwise order.

He compares each current card with the previously examined card. Let x be the previous card's number and y be the current card. Let $g = \gcd(x, y)$, the greatest common divisor of x and y .

The Master remembers the last computed GCD value. If the GCD of the current pair is equal to the previously computed GCD value, and this GCD is greater than 1, he discards the current card y .

This process continues until no more cards can be discarded based on this rule. The Master records the order in which the cards are removed.

- You need to represent the cards 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 card and constructing it with an array is not allowed.

```

1  struct Node {
2      int order, num;
3      struct Node *next, *prev;
4  };
5
6  struct LinkedList {
7      struct Node *head;
8  };

```

Input Format

- The first line contains m ($1 \leq m \leq 10^3$) —the number of cards.
- The second line contains m integers a_1, a_2, \dots, a_m ($1 \leq a_i \leq 10^6$) —the values on the cards.

Output Format

For each test case, output one line:

- First print an integer k —the number of discarded cards.
- Then print k integers —the indices of discarded cards, in order of removal.

Sample Input

```
1 2
2 6
3 12 18 24 30 40 50
4 4
5 7 14 21 28
```

Sample Output

```
1 2 3 4
2 3 3 2 4
```

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 7112056067, your zip file should be DS_2_7112056067.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