2.2 Problem 1

Step 1

Chemical Reactions

In this problem you will need to submit your solution in any of the supported programming languages. The code will be tested on all the prepared tests.

This problem has two versions with different constraints. If your solution passes all the tests of the easy version you will get 200 points. If your solution passes all the tests of the hard version you will get 350 points. In total you can get maximum 550 points. You need to submit correct solutions to **both** versions to get the full score.

The description of the problem follows in the next step. Constraints for each version are also written in the corresponding steps.

Step 2

Note. The problem with '\r\n' should not be the case now. All the tests should have Unix-newlines.

Chemical Reactions

Everybody knows that a huge number of different chemical reactions happen in cells. A reaction takes as an input a set of chemicals (substrates) and converts them to another set (products). Here we consider all reactions to be one-directional. A substrates for a reaction could be chemicals from the environment or products from other reactions.

A scientist John Doe was given a cell and a list of chemicals that are present in the environment at the beginning. He already knows what reactions could happen in the cell. You should help him to understand which chemicals could appear in the cell.

Input Format

The first line contains initial chemicals separated by spaces. There is always at least one initial chemical. Each chemical is represented by a random positive integer that fits in 4-byte integer type. Each of the following lines describes a reaction, one per line. Each reaction is presented as two lists of integers separated by '->': the list of chemicals, separated by '+', that is needed to perform a reaction and the list of chemicals, separated by '+', that are produced as a result of the reaction. Each chemical could be present in each reaction maximum 2 times: one time at the left part and the other time at the right part (for example, a catalyst could appear in both parts).

Constraints for the easy version: a total number of chemicals through all reactions does not exceed 10^3.
Constraints for the hard version: a total number of chemicals through all reactions does not exceed 10^5.

Output Format

The sole line of the output should contain the unordered list of all chemicals that could appear in the cell at any moment of time.

Examples

Sample Input 1

```
4
4+6->1
2->3+5
4->6
6+4->5
```

Sample Output 1:

```
1 4 5 6
```

Sample Input 2

```
1 2
1+2->4
1+2->3
3->4+5
4->4
```

Sample Output 2

```
1 5 2 4 3
```

Limits

Memory limit per test: 256 Megabytes
Time limit per test: 2 seconds

Step 3

Chemical Reactions (Easy)

Constraints for the easy version: a total number of chemicals through all reactions does not exceed 10^3.

Full testset could be downloaded by a link: https://stepik.org/media/attachments/lesson/39302/tests-easy.zip.

Sample Input 1:

```
4
4+6->1
2->3+5
4->6
6+4->5
```

Sample Output 1:

```
1 4 5 6
```

Sample Input 2:

```
1 2
1+2->4
1+2->3
3->4+5
4->4
```

Sample Output 2:

```
1 2 3 4 5
```

To solve this problem please visit https://stepik.org/lesson/39302/step/3

Step 4

Chemical Reactions (Hard)

Constraints for the hard version: a total number of chemicals through all reactions does not exceed 10^5.

Full testset could be downloaded by a link: https://stepik.org/media/attachments/lesson/39302/tests-hard.zip.

Sample Input 1:

```
4
4+6->1
2->3+5
4->6
6+4->5
```

Sample Output 1:

```
1 4 5 6
```

Sample Input 2:

```
1 2
1+2->4
1+2->3
3->4+5
4->4
```

Sample Output 2:

```
1 2 3 4 5
```

To solve this problem please visit https://stepik.org/lesson/39302/step/4

Step 5

Editorial

Easy version

Let us have a set with all the reagents that could be in the cell. We will show how to build it in steps:

1. Populate the set with all initial reagents.
2. While the size of the set changes:
3. Traverse all reactions. For each reaction, we check if all its substrates are in the set. If so, then the reaction could be performed, and we add the products in the set.

Note. If the reaction is performed once, we do not have to perform it the second time. This could be done using a boolean variable.

Note 2. As I've seen the major problem was to not contain reagents in the set. Thus, one reagent could appear multiple times in the answer.

Hard version

We have to store additional information in compare to the previous solution. For each reagent, we know the list of reactions in which it is a substrate. And for each reaction, we store the number of substrates which are not already in the current set.

Aside from this, we will have a set of all the reagents obtained and the queue with reagents to process. The rest of the solution looks like a classical Breadth-First Search Algorithm.

1. We put initial reagents in the set and in the queue.
2. While the queue is not empty:
3. Take the first reagent from the queue.
4. Iterate through the list of reactions in which it appears as a substrate. For each such reaction, we decrease the counter of reagents to be obtained. If this counter becomes zero, this means that we got all the substrates and we could perform a reaction.
5. If the reaction is performed, we iterate through all the products and check its appearance in the set. If it does not appear, then we add this product in the set and in the queue.

If calculated correctly, the complexity of such solution is the total number of chemicals through all the reactions multiplied the complexity of operations on the set.