# Algorithms Advanced with C#: Exam

Please submit your solutions (source code) to all the below-described problems in [Judge](https://judge.softuni.org/Contests/4059).

## 3. E-Shop

You are the owner of a small online store that sells a variety of products. You want to optimize your store's profits by selecting which products to stock based on their popularity and profit margin. Each product has a weight and value (representing its profit margin), and you have limited storage space.

There is a catch: some of the products have a relationship with each other, and if you decide to include one product, you must also include another. However, this rule also is applied recursively, meaning that if you have **Item1-Item2** and **Item2-Item3** pairs, you should take **Item3** if you pick **Item1**.

You must select the optimal products to include in your store to **maximize profits while staying within your storage space**.

### Input

* + The first line of the input contains a positive integer n (1 <= n <= 10^4), representing the number of items.
  + The next **n** lines contain information about the items. Each item is described in the format **"{itemName} {itemWeight} {itemValue}"**.
    - **itemName** is a string representing the name of the item and it will unique.
    - **itemWeight** is an integer representing the weight of the item (1 <= **itemWeight** <= 100)
    - **itemValue** is an integer representing the value of the item (1 <= **itemValue** <= 10^3).
  + On the next line, you will receive a positive integer **p** (1 <= **p** <= 10^4), representing the number of pairs.
  + The next **p** lines contain information about the item pairs. Each item pair is described in the format **"{itemName1} {itemName2}"**, where **itemName1** and **itemName2** are the names of two items that are related to each other.
    - Note that the relationship between items is bidirectional, meaning that if item A is related to item B, then item B is also related to item A.
  + On the last line, you will receive an integer **s** (1 <= **s** <= 10^4), representing the **maximum storage capacity** of the store.

### Output

* + Print the names of all selected items, ordered alphabetically, one item per line.
    - There will be always **only one valid combination** that satisfies all conditions.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3  Item1 3 3  Item2 2 2  Item3 1 4  1  Item1 Item2  4 | Item3 |
| 5  Item1 2 3  Item2 2 3  Item3 2 2  Item4 2 2  Item5 2 20  2  Item1 Item3  Item1 Item4  6 | Item2  Item5 |