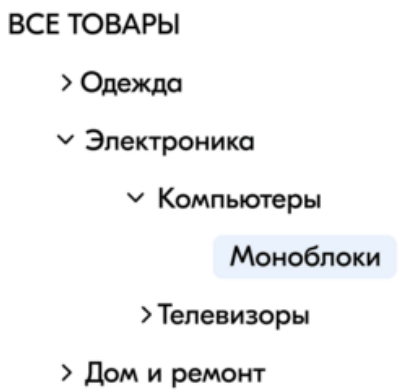
F. JSON категории (25 баллов)

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

На маркетплейсе у каждого товара есть категория. При этом, у некоторых категорий есть дочерние категории. Для удобной навигации по маркетплейсу покупатели могут пользоваться деревом категорий.



Ваша задача — построить дерево категорий. Дана информация об отношениях родительских и дочерних категорий в виде JSON-массива. Каждый элемент массива является словарем, с полями `name` (название категории), `id` (числовой идентификатор категории) и `parent` (числовой идентификатор родительской категории). Известно, что корневая категория имеет нулевой идентификатор и не имеет идентификатора родительской категории.
По данной информации постройте дерево категорий в виде JSON-словаря. Словарь для каждой категории должен иметь поля `name`, `id` и массив `next`, состоящий из таких же словарей для дочерних категорий.

Входные данные

Каждый тест состоит из нескольких наборов входных данных. Первая строка содержит целое число t ($1 \le t \le 100$) — количество наборов входных данных. Далее следует описание наборов входных данных.

Первая строка каждого набора входных данных содержит целое число n ($1 \le n \le 1000$) — количество строк с описанием JSON-массива категорий.

Следующие n строк содержат описание JSON-массива категорий. Все числовые идентификаторы категорий являются целыми числами и удовлетворяют условию $0 \le id \le 10^9$. Все имена категорий непустые, состоят из строчных латинских букв и имеют длину не больше 20. В описании могут быть символы пробела и табуляции.

Гарантируется, что каждый набор входных данных содержит корневую категорию и не более 400 категорий.

Гарантируется, что размер входных данных не превосходит 10Мб.

Выходные данные

Выведите JSON-массив из t элементов. i-й элемент массива является словарем с описанием дерева категорий для i-го набора входных данных. При проверке ответа пробелы, табы и переносы строки не учитываются(кроме таковых в json полях). Порядок полей в словаре и порядок дочерних категорий в массиве `next` не учитывается. Если у категории нет дочерних категорий, ключ `next` может отсутствовать, или соответствовать пустому массиву.

Примеры

входные данные Скопировать

```
2
21
[
   {
      "id":0,
      "name":"all"
```

```
   },
   {
      "id":1,
      "name":"clothes",
      "parent":0
   },
   {
      "id":2,
      "name":"shoes",
      "parent":0
   },
   {
      "id":55,
      "name":"sneakers",
      "parent":2
   }
]
6
 [ {"parent":   0,"id":100,  "name":
 "x"},{

"name":"x","id":0}

]
```

выходные данные Скопировать

```
[{
        "id": 0,
        "name": "all",
        "next": [{
                "id": 1,
                "name": "clothes",
                "next": []
        }, {
                "id": 2,
                "name": "shoes",
                "next": [{
                        "id": 55,
                        "name": "sneakers"
                }]
        }]
},
{"name":"x","id":0,"next":[{"id":100,"name":"x"}]}
]
```

входные данные Скопировать

```
1
9
[
 {"name": "everything", "id": 0},
 {"name": "clothes", "id": 1, "parent": 0},
 {"name": "electronics", "id": 2, "parent": 0},
 {"name": "computers", "id": 4, "parent": 2},
 {"name": "aio", "id": 3, "parent": 4},
 {"name": "tv", "id": 5, "parent": 2},
 {"name": "house", "id": 6, "parent": 0}
]
```

выходные данные Скопировать

```
[
{"id":0,"name":"everything","next":[{"id":1,"name":"clothes","next":[]},{"id":2,"name":"electronics","next":
[{"id":4,"name":"computers","next":[{"id":3,"name":"aio","next":[]}]},{"id":5,"name":"tv","next":[]}]},
{"id":6,"name":"house","next":[]}]}
]
```

Примечание

Любые внешние библиотеки использовать нельзя.

- Для работы с JSON в языке C# можно пользоваться библиотекой System.Text.Json. Возможно, вам понадобится увеличить максимальную глубину сериализации/десериализации JSON (документация).
- Для работы с JSON в языке Go можно пользоваться библиотекой encoding/json.
- Для работы с JSON в языке Python можно пользоваться библиотекой json.