**Опять JSON’ы перекладывать...**

Ограничение памяти

256.0 Мб

Ограничение времени

2 с

Ввод

стандартный ввод или input.txt

Вывод

стандартный вывод или output.txt

Всем хотя бы раз в жизни приходилось перекладывать JSON. Вот и для нового проекта под названием "Единое хранилище" необходимо переложить магазинные фиды. Для размещения на Яндекс.Маркете магазины передают товары из своего ассортимента в JSON-файлах. Одно товарное предложение описывается так:

{

"offer\_id": <string>,

"market\_sku": <int>,

"price": <int>

}

где *offer*\_*id* - уникальный среди всех фидов магазина идентификатор предложения, *market*\_*sku* - идентификатор товара на Яндекс.Маркете, *price* - стоимость товара.

Весь фид магазина представляет собой JSON и выглядит так:

{

"offers": [<offer>, <offer>, ...]

}

Вас попросили написать программу, которая объединит все фиды одного магазина в единый фид и отсортирует товарные предложения в порядке неубывания их стоимостей, а при их равенстве - по *offer*\_*id*.

**Формат ввода**

В первой строке входных данных содержится целое число n*n* - количество фидов магазина (1≤*n*≤200). Следующие *n* строк содержат по одному магазинному фиду на строку. Гарантируется, что строка - валидный JSON и удовлетворяет формату фида. В одном фиде не больше 200 товарных предложений. *offer*\_*id* состоит из строчных и заглавных букв латинского алфавита и цифр, 1≤∣*offer*\_*id*∣≤10, 1≤*market*\_*sku*≤231−1, 1≤*price*≤106.

**Формат вывода**

Выходной поток должен содержать один JSON-документ, удовлетворяющий формату товарного фида. Количество строк в выходном файле и табуляция не имеют значения.

**Пример**

**Ввод**

2

{"offers": [{"offer\_id": "offer1", "market\_sku": 10846332, "price": 1490}, {"offer\_id": "offer2", "market\_sku": 682644, "price": 499}]}

{"offers": [{"offer\_id": "offer3", "market\_sku": 832784, "price": 14000}]}

**Вывод**

{"offers":[{"market\_sku":682644,"offer\_id":"offer2","price":499},{"market\_sku":10846332,"offer\_id":"offer1","price":1490},{"market\_sku":832784,"offer\_id":"offer3","price":14000}]}

**Примечание**

Для решений на языке **python** доступны все стандартные библиотеки, включая json.

Для решений на языке **Java** доступна библиотека [json-simple](https://code.google.com/archive/p/json-simple/) версии 1.1.1. Соответствующие import'ы могут выглядеть так:

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import org.json.simple.parser.ParseException;

Для решений на языке **C++** доступна библиотека [nlohmann/json](https://github.com/nlohmann/json/releases/tag/v3.8.0) v3.8.0. Соответствующие include выглядят так:

#include "json.hpp"

Для решений на **golang** доступны все стандартные пакеты, включая encoding/json, sort и другие.