

Лабораторная работа 1.

Контейнеры map и set.

Часть 1.

Выполните следующие задачи по ссылке:

<https://informatics.msk.ru/mod/statements/view.php?id=81066#1>

Часть 2.

Задача 1. Хеширование с удалением (2 балла)

Используйте `unordered_set` для реализации множества строк.

Хранимые строки – непустые последовательности длиной не более 10 символов, состоящие из строчных латинских букв. Структура данных должна поддерживать операции добавления строки в множество, удаления строки из множества и проверки принадлежности данной строки множеству. Максимальное количество элементов в хранимом множестве не превосходит 10^6 .

Формат ввода

Каждая строка входных данных задает одну операцию над множеством.

Запись операции состоит из типа операции и следующей за ним через пробел строки, над которой проводится операция. Тип операции – один из трех символов:

+ означает добавление данной строки в множество;

- означает удаление строки из множества;

? означает проверку принадлежности данной строки множеству.

Общее количество операций во входном файле не превосходит 10^6 . Список операций завершается строкой, в которой записан один символ # – признак конца входных данных.

При добавлении элемента в множество НЕ ГАРАНТИРУЕТСЯ, что он отсутствует в этом множестве. При удалении элемента из множества НЕ ГАРАНТИРУЕТСЯ, что он присутствует в этом множестве.

Формат вывода

Программа должна вывести для каждой операции типа ? одну из двух строк YES или NO, в зависимости от того, встречается ли данное слово в нашем множестве.

Примеры.

ВВОД	ВЫВОД
+ hello	YES
+ bye	NO
? bye	YES

- bye ? bye ? hello #	
? qwertyuiop #	NO
? asdfghjklz + asdfghjklz ? asdfghjklz #	NO YES

Задача 2. (2 балла)

Профессор написал научную книгу и составил для неё предметный указатель. Это список ключевых слов, для каждого из которых указана страница, на которой это слово встречается. Теперь профессор хочет для каждой страницы выписать в алфавитном порядке все ключевые слова, которые на эту страницу попали (если такие вообще есть). Помогите профессору решить эту задачу.

Формат ввода

Сначала задано натуральное число n , не превосходящее 1000 — количество слов, которое требуется обработать. Далее идут n строк. В каждой строке сначала записано ключевое слово. Затем идёт натуральное число, также не превосходящее 1000, — номер страницы. Ключевые слова состоят из латинских букв, не бывают пустым и по длине не превосходят 16 символов. Слова в списке, конечно, могут повторяться.

Формат вывода

Выпишите в порядке возрастания все страницы, на которых присутствуют ключевые слова. После каждого номера страницы через пробел выпишите в алфавитном порядке сами эти слова. Если на какой-то странице слово встретилось несколько раз, то повторять его не нужно. Завершающего пробела в конце строк быть не должно.

Примеры.

ВВОД	ВЫВОД
5 derivative 10 function 2 function 10 function 10 limit 7	2 function 7 limit 10 derivative function

Задача 3. word2vec (2 балла)

Известная утилита word2vec, основанная на нейронной сети, умеет приписывать каждому слову из словаря некоторый числовой вектор размерности N . При этом оказывается, что семантически «близким» словам соответствуют «близкие» векторы. Эта утилита находит широкое применение в задачах обработки естественного языка и машинного обучения.

В качестве меры «близости» между словами обычно выбирают угол между их векторами, косинус которого вычисляют по известной формуле

$$\cos \alpha = \frac{\langle u, v \rangle}{\sqrt{\langle u, u \rangle} \sqrt{\langle v, v \rangle}}$$

Однако часто векторы слов нормируют (чтобы их длина стала равна константе), и тогда в качестве меры близости рассматривают просто скалярное произведение

$$\langle u, v \rangle = \sum_{i=1}^N u_i v_i$$

Чем это скалярное произведение больше, тем семантически «ближе» слова друг к другу.

Вам дан словарь из M слов, каждому из которых сопоставлен N -мерный вектор. Гарантируется, что все слова различны. Требуется найти в этом списке ближайшие слова к первому слову (кроме самого этого первого слова), используя в качестве метрики близости просто скалярное произведение.

Формат ввода

В первой строке через пробел записано 2 целых числа — M ($2 \leq M \leq 100$) и N ($3 \leq N \leq 1000$) — количество слов в словаре и размерность векторов соответственно.

Далее следует M строк, каждая из которых описывает очередное слово.

Описание состоит из самого слова, длиной не более 15 символов, состоящего из строчных латинских букв, и набора из N целых чисел, разделенных пробелами. Компоненты вектора по модулю не превосходят 1000.

Формат вывода

Выпишите слова (по одному на строке), скалярное произведение векторов которых с вектором первого слова максимально. При этом скалярное произведение первого слова с самим собой учитывать не нужно. Если таких слов несколько, то их следует вывести в том же порядке, в котором они были даны на входе.

ВВОД	ВЫВОД
4 3 sweden 4 6 3 queen 0 2 2	norway

norway 4 7 5 king -1 3 2	
-----------------------------	--

Задача 4. (2 балла)

Рассмотрим последовательность целых чисел длины n . По ней с шагом 1 двигается «окно» длины k , то есть сначала в «окне» видны первые k чисел, на следующем шаге в «окне» уже будут находиться k чисел, начиная со второго, и так далее до конца последовательности. Требуется для каждого положения «окна» определить минимум в нём.

Формат ввода

В первой строке входных данных содержатся два натуральных числа n и k ($n \leq 150000$, $k \leq 10000$, $k \leq n$) — длины последовательности и «окна», соответственно. На следующей строке находятся n чисел — сама последовательность.

Формат вывода

Выходные данные должны содержать $n - k + 1$ строк — минимумы для каждого положения «окна».

ВВОД	ВЫВОД
7 3 1 3 2 4 5 3 1	1 2 2 3 1

Задача 5. (2 балла)

Реализуйте систему хранения автобусных маршрутов. Вам нужно обрабатывать следующие запросы:

- **NEW_BUS** *bus stop_count stop1 stop2 ...* — добавить маршрут автобуса с названием *bus* и *stop_count* остановками с названиями *stop1*, *stop2*, ...
- **BUSES_FOR_STOP** *stop* — вывести названия всех маршрутов автобуса, проходящих через остановку *stop*.
- **STOPS_FOR_BUS** *bus* — вывести названия всех остановок маршрута *bus* со списком автобусов, на которые можно пересесть на каждой из остановок.
- **ALL_BUSES** — вывести список всех маршрутов с остановками.

Формат ввода

В первой строке ввода содержится количество запросов Q , затем в Q строках следуют описания запросов.

Гарантируется, что все названия маршрутов и остановок состоят лишь из латинских букв, цифр и знаков подчёркивания.

Для каждого запроса **NEW_BUS** *bus stop_count stop1 stop2 ...* гарантируется, что маршрут *bus* отсутствует, количество остановок больше 0, а после числа *stop_count* следует именно такое количество названий остановок, причём все названия в каждом списке различны.

Формат вывода

Для каждого запроса, кроме **NEW_BUS**, выведите соответствующий ответ на него:

- На запрос **BUSES_FOR_STOP** *stop* выведите через пробел список автобусов, проезжающих через эту остановку, в том порядке, в котором они создавались командами **NEW_BUS**. Если остановка *stop* не существует, выведите **No stop**.
- На запрос **STOPS_FOR_BUS** *bus* выведите описания остановок маршрута *bus* в отдельных строках в том порядке, в котором они были заданы в соответствующей команде **NEW_BUS**. Описание каждой остановки *stop* должно иметь вид **Stop stop: bus1 bus2 ...**, где *bus1 bus2 ...* — список автобусов, проезжающих через остановку *stop*, в порядке, в котором они создавались командами **NEW_BUS**, за исключением исходного маршрута *bus*. Если через остановку *stop* не проезжает ни один автобус, кроме *bus*, вместо списка автобусов для неё выведите **no interchange**. Если маршрут *bus* не существует, выведите **No bus**.
- На запрос **ALL_BUSES** выведите описания всех автобусов в алфавитном порядке. Описание каждого маршрута *bus* должно иметь вид **Bus bus: stop1 stop2 ...**, где *stop1 stop2 ...* — список остановок автобуса *bus* в порядке, в котором они были заданы в соответствующей команде **NEW_BUS**. Если автобусы отсутствуют, выведите **No buses**.

Предупреждение

Условие задачи выше содержит много важных деталей. Если вы не уверены в том, что не упустили ни одной, перечитайте условие ещё раз.

ВВОД	ВЫВОД
10 ALL_BUSES BUSES_FOR_STOP Marushkino STOPS_FOR_BUS 32K NEW_BUS 32 3 Tolstopaltsevo Marushkino Vnukovo	No buses No stop No bus 32 32K Stop Vnukovo: 32 32K 950 Stop Moskovsky: no interchange

NEW_BUS 32K 6 Tolstopaltsevo Marushkino Vnukovo Peredelkino Solntsevo Skolkovo BUSES_FOR_STOP Vnukovo NEW_BUS 950 6 Kokoshkino Marushkino Vnukovo Peredelkino Solntsevo Tropanyovo NEW_BUS 272 4 Vnukovo Moskovsky Rumyantsevo Tropanyovo STOPS_FOR_BUS 272 ALL_BUSES	Stop Rumyantsevo: no interchange Stop Tropanyovo: 950 Bus 272: Vnukovo Moskovsky Rumyantsevo Tropanyovo Bus 32: Tolstopaltsevo Marushkino Vnukovo Bus 32K: Tolstopaltsevo Marushkino Vnukovo Peredelkino Solntsevo Skolkovo Bus 950: Kokoshkino Marushkino Vnukovo Peredelkino Solntsevo Tropanyovo
--	---

Задача 6. (2 балла)

В этой задаче вам нужно присваивать номера автобусным маршрутам.

А именно, для каждого маршрута, заданного множеством названий остановок, нужно либо выдать новый номер (первому маршруту — 1, второму — 2 и т. д.), либо вернуть номер существующего маршрута, которому соответствует такое множество остановок.

В отличие от задачи «Автобусные остановки — 2», наборы остановок, которые можно получить друг из друга перестановкой элементов или добавлением/удалением повторяющихся, следует считать одинаковыми.

Формат ввода

Сначала вводится количество запросов Q , затем Q описаний запросов.

Каждый запрос представляет собой положительное количество остановок N , за которым следуют разделённые пробелом N названий остановок соответствующего маршрута (не обязательно различных). Названия остановок состоят лишь из латинских букв и символов подчёркивания.

Формат вывода

Выведите ответ на каждый запрос в отдельной строке.

Если маршрут с данным набором остановок уже существует, в ответ на соответствующий запрос выведите **Already exists for i** , где i — номер маршрута с таким набором остановок. В противном случае нужно выделить введённому набору остановок новый номер i и вывести его в формате **New bus i** .

ВВОД	ВЫВОД
5 2 Marushkino Kokoshkino	New bus 1 New bus 2

1 Kokoshkino	Already exists for 1
2 Marushkino Kokoshkino	Already exists for 1
2 Kokoshkino Marushkino	Already exists for 2
2 Kokoshkino Kokoshkino	