

## Задача А. Корень кубического уравнения

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.25 секунд  
Ограничение по памяти: 256 мегабайт

Дано кубическое уравнение  $ax^3 + bx^2 + cx + d = 0$  ( $a \neq 0$ ). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

### Формат входных данных

Во входном файле через пробел записаны четыре целых числа:  $-1000 \leq a, b, c, d \leq 1000$ .

### Формат выходных данных

Выведите единственный корень уравнения с точностью не менее 5 знаков после десятичной точки.

### Примеры

стандартный ввод	стандартный вывод
1 -3 3 -1	1.0000003749
-1 -6 -12 -7	-1.0000000111

## Задача В. Введите одностороннее движение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В Тридевятом Царстве было  $N$  городов, некоторые из которых были соединены дорогами. К сожалению, в последнее время добраться из одного города в другой стало очень сложно из-за возникших автомобильных пробок. В целях борьбы с пробками было решено все дороги сделать односторонними, т.е. разрешить проезд по каждой дороге только в одном направлении. При этом требуется, чтобы по-прежнему можно было из любого города попасть в любой другой.

### Формат входных данных

Во входном файле записано сначала число  $N$  — количество городов ( $1 \leq N \leq 1000$ ). Затем записано число  $M$  — количество дорог ( $1 \leq M \leq 100000$ ). Далее идет  $M$  пар чисел, задающих дороги (каждая дорога описывается номерами городов, которые она соединяет). Не бывает дорог из некоторого города в тот же город. Между двумя городами может быть несколько дорог. Гарантируется, что до введения одностороннего движения можно было попасть из любого города в любой другой.

### Формат выходных данных

В выходной файл нужно выдать  $M$  пар чисел, соответствующих дорогам (дороги должны быть выданы в том же порядке, в котором они заданы во входном файле). Для каждой дороги сначала должен быть записан номер города, из которого по ней можно будет уехать после введения одностороннего движения, а затем — номер города, куда эта дорога ведет.

Если ввести одностороннее движение так, чтобы можно было из любого города попасть в любой другой, нельзя, выходной файл должен содержать одно число 0.

### Пример

стандартный ввод	стандартный вывод
4	1 2
6	2 1
1 2	2 3
1 2	4 2
2 3	3 4
2 4	4 1
4 3	
1 4	

## Задача С. Потерянные пробелы

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Преподаватель Алексей много времени проводит, просматривая посылки учеников. Он считает, что если подробно описать причину, по которой он отклонил посылку, то ученики больше не будут допускать такие ошибки. Также он любит писать комментарии на эстонском, чтобы ученики параллельно с решением задач учились иностранному языку.

Однажды он настолько увлёкся написанием комментария, что не заметил, как сломалась клавиша пробел. Дело в том, что Алексей ещё не научился печатать вслепую, поэтому не смотрел на экран и не увидел, что пробел не набирается. Более того, он был настолько увлечён, что писал комментарий одним предложением без знаков препинания и прописных букв.

К счастью, Алексей при написании комментария пользовался русско-эстонским словарём и не допускал опечаток при наборе текста. Также он использовал только символы латинского алфавита, поскольку не знает, как набирать разные закорючки рядом с буквами на своём ноутбуке.

Помогите Алексею вернуть пробелы в комментарий, который он так долго писал. Он обрадуется любому варианту расстановки пробелов тексте, лишь бы полученный текст состоял из корректных слов.

### Формат входных данных

Первая строка содержит одно целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество слов в словаре.

Каждая из следующих  $n$  строк содержит очередное эстонское слово  $s_i$  ( $1 \leq |s_i|; \sum |s_i| \leq 10^5$ ). Каждое слово состоит только из строчных букв латинского алфавита. Все слова различны.

Последняя строка содержит  $t$  ( $1 \leq |t| \leq 10^5$ ) — текст без пробелов, состоящий только из строчных букв латинского алфавита.

### Формат выходных данных

Выведите текст, полученный расстановкой пробелов в заданном тексте. Каждое слово отделенное пробелом должно присутствовать в заданном словаре.

Гарантируется, что существует хотя бы одно корректное разбиение текста на слова. Если существует несколько способов расставить пробелы в тексте, то разрешается вывести любое из них.

### Примеры

стандартный ввод	стандартный вывод
2 kood hea heakood	hea kood
9 kood suboptimaalne dubleerimist transliteratsioon kasuta saa ei muutujad nimetada eisaanimetadamuutujadtransliteratsioon	ei saa nimetada muutujad transliteratsioon

## Задача D. Защищенное соединение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В свете недавних новостей о прослушке каналов связи, два непримиримых интернет-гиганта Урагании «Laim.UR» и «Xenda» решили подписать соглашение об установлении защищенного канала связи между дата-центрами друг друга. В Урагании  $n$  городов, но, к сожалению, ни в одном городе нет дата-центров обоих гигантов. Поэтому для формирования защищенного канала придется прокладывать междугородние линии связи.

Специалисты компаний определили  $m$  пар городов, которые можно соединить, проложив сегмент канала связи, и оценили стоимость создания такого сегмента для каждой из этих пар.

Результирующий канал может состоять из нескольких сегментов. Он должен начинаться в одном из городов, где находится дата-центр первой компании, может проходить через промежуточные города и должен заканчиваться в городе, где находится дата-центр второй компании.

Теперь необходимо определить минимальную стоимость защищенного канала, соединяющего два дата-центра компаний.

### Формат входных данных

В первой строке находятся целые числа  $n$  и  $m$  ( $2 \leq n \leq 5000$ ,  $1 \leq m \leq 10^5$ ) — количество городов и количество пар городов, которые можно соединить сегментом канала связи.

Во второй строке находятся  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 2$ ). Если  $a_i = 0$ , то в  $i$ -м городе нет дата-центра ни одного из гигантов. Если  $a_i = 1$ , то в  $i$ -м городе есть дата-центр «Laim.UR», а если  $a_i = 2$ , то в  $i$ -м городе находится дата-центр «Xenda». Гарантируется, что среди этих чисел есть как минимум одна единица и одна двойка.

В каждой из следующих  $m$  строк находится по три целых числа —  $s_i$ ,  $t_i$  и  $c_i$ , которые означают, что города  $s_i$  и  $t_i$  ( $1 \leq s_i, t_i \leq n$ ,  $s_i \neq t_i$ ) можно соединить сегментом канала связи стоимостью  $c_i$  ( $1 \leq c_i \leq 10^5$ ). Каждую пару городов можно соединить не более чем одним сегментом канала.

### Формат выходных данных

Если соединить защищенным каналом связи два дата-центра разных интернет-гигантов возможно, то выведите в выходной файл три числа:  $x$ ,  $y$  и  $d$ , означающие, что между городами  $x$  и  $y$  возможно провести канал связи суммарной стоимостью  $d$ . В городе  $x$  должен находиться дата-центр «Laim.UR», в городе  $y$  — дата-центр «Xenda». Если существует несколько оптимальных ответов, выведите любой. Если провести искомый канал невозможно, выведите -1.

### Примеры

стандартный ввод	стандартный вывод
6 7 1 0 1 2 2 0 1 3 3 1 2 4 2 3 3 2 4 2 1 6 5 3 5 6 5 6 1	3 4 5
4 2 1 0 0 2 1 3 3 2 4 2	-1

## Задача E. Type Printer

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Вам нужно напечатать  $N$  слов на *Movable Type Printer*. *Movable Type Printers* — это старые принтеры, для работы которых требуется ставить маленькие металлические кусочки (каждый из кусочков содержит одну букву) в определенном порядке, образуя таким образом слова. Потом все они вдавливаются в лист бумаги. Таким образом печатается одно слово. Ваш принтер позволяет делать следующие операции:

- Добавить букву в конец слова, находящегося сейчас на принтере.
- Удалить последнюю букву из слова, находящегося сейчас на принтере. Эту операцию можно делать, только если слово содержит хотя бы одну букву.
- Напечатать слово, находящееся на принтере (при этом слово никуда не исчезает, можно печатать его ещё раз и ещё раз).

Изначально на принтере содержится пустое слово. В конце печати на принтере можно оставить непустое слово. Слова, которые вам даны, вы можете печатать в произвольном порядке.

Каждая из трёх операций занимает одну единицу времени. Вам нужно найти последовательность операций, которая печатает данные  $N$  слов за минимальное время. Если минимальных последовательностей несколько, выведите любую.

### Формат входных данных

В первой строке содержится число  $N$  ( $1 \leq N \leq 25\,000$ ). В следующих  $N$  строках содержатся слова (по одному на строке), состоящие из маленьких букв латинского алфавита. Длина каждого слова не превышает 20. Все слова различны.

### Формат выходных данных

Ваша программа должна вывести следующие данные:

- На первой строке число  $M$  — число операций.
- На следующих  $M$  строках по одному символу — описание операций. Каждая операция описывается одним символом:
  - Добавление символа обозначается собственно символом.
  - Удаление символа обозначается символом «-» (минус, ASCII-код 45).
  - Операция «напечатать текущее слово» обозначается символом «P» (заглавная латинская буква P).

## Пример

стандартный ввод	стандартный вывод
3 print the poem	20 t h e p - - - p o e m p - - - r i n t p