

## Задача А. Граф

Имя входного файла: `graph.in`  
Имя выходного файла: `graph.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Много лет назад Альфу на день рождения подарили неориентированный граф. Недавно, разбирая вещи, он нашел его, и сильно удивился. Поиграв с ним некоторое время, ему стало интересно: сколько в этом графе есть пар вершин, между которыми нет ребра, а если его провести, в графе появится ровно один новый вершинно-простой цикл?

### Формат входного файла

В первой строке входного файла заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 10^5$ ) — число вершин и ребер в графе соответственно.

В следующих  $m$  строках заданы пары целых чисел  $u, v$  ( $1 \leq u, v \leq n$ ) — ребра графа.

### Формат выходного файла

В единственной строке выходного файла выведите искомое число пар вершин.

### Пример

<code>graph.in</code>	<code>graph.out</code>
5 4 1 2 2 3 3 4 4 5	6
5 5 1 2 2 3 1 3 3 4 4 5	1

## Задача В. Любимая строка

Имя входного файла: `string.in`  
Имя выходного файла: `string.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В очередной раз копаясь в гараже Вилли, Альф нашел строку, которую, судя по всему, хозяин гаража мастерил с особым усердием. Любознательный пришелец, как обычно, не растерялся и решил поиграть с этой строкой. Он разломал ее на  $k$  частей одинаковой длины и разбросал по всему гаражу. Вилли, увидев что случилось с его строкой, пришел в ярость. Свою злость он выместил на Альфе, приказав ему склеить строку обратно. К счастью для пришельца, у Вилли нашелся чертеж этой строки.

Помогите Альфу воссоздать строку по кускам и чертежу!

### Формат входного файла

В первой строке входного файла даны два числа  $n$  и  $m$  ( $1 \leq n, m \leq 10^6$ ) — длина строки и количество кусков соответственно. Во второй строке входного файла дана строка  $s$  ( $1 \leq |s| \leq 10^6$ ) — строка, которую необходимо получить. В  $i$ -й из следующих  $m$  строк дана строка  $t_i$  — описание  $i$ -го куска исходной строки. Гарантируется, что  $n$  делится на  $m$  и из данных кусков можно составить исходную строку.

### Формат выходного файла

В единственной строке выходного файла выведите  $m$  различных целых чисел  $a_i$  ( $1 \leq a_i \leq m$ ), таких, что если заменить число на  $i$ -й позиции куском с номером  $a_i$  и склеить получившиеся куски, получится исходная строка.

Если ответов несколько, выведите любой.

### Пример

<code>string.in</code>	<code>string.out</code>
12 3 cabacaqwerty erty caba caqw	2 3 1

## Задача С. Домашнее задание

Имя входного файла: `tree.in`  
Имя выходного файла: `tree.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Линн уже взрослая, скоро она пойдет в колледж. Но поступить туда не так просто. Она решила прорешивать задачи старых лет, чтобы лучше подготовиться.

Одним из заданий на вступительной олимпиаде прошлого года была следующая задача: дано дерево, в котором в  $i$ -й вершине записано число  $a_i$ . Все, что требовалось от участников олимпиады — посчитать сумму

$$\sum_{i=1}^n \sum_{j=i}^n (\max(i \dots j) - \min(i \dots j)) \cdot \text{len}(i \dots j)$$

Здесь  $\max(i \dots j)$  — максимальное из значений  $a_k$  для всех вершин  $k$  на пути из  $i$  в  $j$ ,  $\min(i \dots j)$  — минимальное из тех же значений,  $\text{len}(i \dots j)$  — число вершин на пути из  $i$  в  $j$ .

Линн уже который час пытается решить эту задачу, но все безуспешно. Помогите ей!

### Формат входного файла

В первой строке входного файла записано целое число  $n$  ( $1 \leq n \leq 5 \cdot 10^4$ ) — число вершин в дереве. Во второй строке заданы  $n$  целых чисел  $a_i$  ( $0 \leq a_i \leq 10^5$ ) — числа, записанные в вершинах. В следующих  $n - 1$  строках заданы пары целых чисел  $u, v$  ( $1 \leq u, v \leq n$ ) — ребра дерева.

### Формат выходного файла

В единственной строке выходного файла выведите значение искомой суммы.

### Пример

tree.in	tree.out
5 1 2 3 4 5 1 2 2 3 3 4 4 5	70
4 0 1 1 1 1 2 1 3 1 4	15

## Задача D. Ученье — свет, а неученье — тьма

Имя входного файла: `divisors.in`  
Имя выходного файла: `divisors.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 512 мегабайт

Последнее время Альф вообще ничего не делает. Даже ничего не сломал за последнюю неделю. Линн решила, что ему пора прекратить бездельничать и начать учиться.

Начать обучение она решила с несложной задачи: дан массив натуральных чисел. С массивом производятся две операции:

- `0 i x` — присвоить  $i$ -му элементу массива значение  $x$ ;
- `1 l r` — вывести количество делителей у произведения чисел на отрезке с  $l$  по  $r$ . Это число может быть довольно большим, поэтому ответ требуется вывести по модулю  $10^9 + 7$ .

Альф тайно влюблен в Линн, поэтому не может ей отказать. Но и работать он не хочет. Поэтому он попросил вас в тайне от Линн написать программу, которая решает данную задачу.

### Формат входного файла

В первой строке входного файла дано число  $n$  ( $1 \leq n \leq 5 \cdot 10^4$ ) — количество элементов массива.

Во второй строке дано  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^4$ ) — элементы массива.

В третьей строке дано число  $q$  ( $1 \leq q \leq 10^4$ ) — количество запросов.

В каждой из следующих  $q$  строк записано по три числа. Если первое из этих чисел равно 0, то это запрос обновления элемента массива, если же первое число равно 1, то это запрос на нахождение количества делителей у произведения чисел на отрезке по модулю  $10^9 + 7$ .

Гарантируется, что во всех запросах  $1 \leq i, l, r \leq n; 1 \leq x \leq 10^4$ .

### Формат выходного файла

На каждый запрос нахождения количества делителей выведите в отдельной строке выходного файла ответ — количество делителей у произведения всех чисел на данном отрезке по модулю  $10^9 + 7$ .

### Пример

<code>divisors.in</code>	<code>divisors.out</code>
5	12
2 3 4 5 6	6
6	6
1 2 4	12
1 2 3	
0 1 1	
0 4 7	
1 1 3	
1 1 4	

## Задача Е. Новое развлечение

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Альф и Лаки очень много времени проводят вместе. Сегодня Альф придумал новую игру и сразу же рассказал Лаки ее правила.

В игре участвуют два игрока, которые ходят по очереди. Изначально на доску выписываются два числа  $a$  и  $b$ . Своим ходом игрок может либо взять любое из этих двух чисел и поделить его на собственный делитель, либо оба числа поделить на их общий собственный делитель. В данной задаче под собственным делителем числа  $x$  будем понимать такое число  $y$ , что  $x$  делится на  $y$  и  $y$  строго больше единицы.

Проигрывает тот, кто не может сделать очередной ход.

Право первого хода Альф дал Лаки, который уже давно хочет поквитаться с Альфом за все его издевательства над ним, поэтому непременно хочет выиграть. Помогите Лаки — скажите, может ли он выиграть, если и он, и Альф играют оптимально.

### Формат входного файла

В первой и единственной строке входного файла содержатся два числа  $a, b$  — числа, изначально выписанные на доску ( $1 \leq a, b \leq 10^9$ ).

### Формат выходного файла

В единственной строке выходного файла выведите `Lucky wins`, если Лаки может выиграть, и `Alf wins` — иначе.

### Пример

<code>game.in</code>	<code>game.out</code>
2 2	Lucky wins
2 4	Alf wins

## Задача F. Рутинная работа

Имя входного файла: `stacks.in`  
Имя выходного файла: `stacks.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вчера Альф в очередной раз провинился — опять чуть не съел Лаки. На этот раз Вилли решил проучить Альфа.

В кладовке у него как раз завалились  $n + 1$  очередь и  $n$  стеков. Причем в одной из очередей также находилось  $2^n$  различных целых чисел от 1 до  $2^n$ . Вилли выложил перед Альфом все очереди и стеки в чередующемся порядке — сначала идет очередь, потом стек, потом опять очередь, и так далее, последней Вилли выложил очередь. Причем очередь с числами оказалась первой.

Все, что Альф может делать с этими стеками и очередями — вынуть из структуры данных номер  $i$  первое число (для стека это число на вершине, для очереди — число, находящееся в голове) и добавить его в структуру данных номер  $i + 1$  (в случае стека число добавится на его вершину, в случае очереди — в хвост). Теперь Вилли просит Альфа отсортировать все числа из первой очереди — сделать несколько операций над этими структурами данных, чтобы все числа находились в последней очереди, а также располагались там в возрастающем порядке. То есть должно быть выполнено, что в голове очереди находится число 1, в хвосте очереди —  $2^n$ , а между ними числа должны быть отсортированы.

Вилли утверждает, что отсортировать числа гарантированно можно за  $2 \cdot 2^n \cdot n$  операций. Помогите Альфу понять, как нужно применять эти операции, чтобы выполнить задание Вилли.

### Формат входного файла

В первой строке входного файла дано целое число  $n$  ( $1 \leq n \leq 15$ ) — число стеков. Во второй строке входного файла дано  $2^n$  различных чисел  $a_i$  ( $1 \leq a_i \leq 2^n$ ) — исходный набор чисел.

### Формат выходного файла

В первой и единственной строке выходного файла выведите  $2 \cdot 2^n \cdot n$  чисел  $b_i$  ( $1 \leq b_i \leq 2n$ ) — номер структуры данных, из которой извлекается число на  $i$ -м шаге вашего решения. Данная структура не должна быть пуста.

### Пример

<code>stacks.in</code>	<code>stacks.out</code>
1	1 2 1 2
1 2	
1	1 1 2 2
2 1	

### Комментарий

<https://ru.wikipedia.org/wiki/Стек>

### Комментарий

[https://ru.wikipedia.org/wiki/Очередь\\_\(программирование\)](https://ru.wikipedia.org/wiki/Очередь_(программирование))

## Задача G. Ломать — не строить

Имя входного файла: `segments.in`  
Имя выходного файла: `segments.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Все ушли из дома, и теперь Альфу нечем заняться. Но тут он увидел на стене карту Лос-Анджелеса и решил ее поизучать. Больше всего Альфу интересны автомобильные дороги, ведь на его родной планете Мелмак их не было — все автомобили там уже давным-давно летают.

Дорожная система Лос-Анджелеса представляет собой набор из  $n$  точек общего положения на плоскости, между которыми проведены отрезки — двусторонние дороги. Причем из каждой точки проведено ровно два отрезка — это порядком удивило Альфа. Перерисовав дорожную систему себе на листок, Альф решил удалить из нее какие-то дороги, ведь он так любит все ломать! Но он не хочет удалять любые дороги. Он хочет у каждой точки удалить ровно одну дорогу, исходящую из нее, так, чтобы после этого никакие две дороги не пересекались. Альф считает, что дороги пересекаются, если у них есть общая точка, отличная от их концов.

Альф очень любопытный, помогите ему узнать, можно ли сделать то, что он хочет.

### Формат входного файла

В первой строке входного файла содержится число  $n$  ( $3 \leq n \leq 500$ ) — количество точек на плоскости.

В  $i$ -й из следующих  $n$  строк содержатся два целых числа  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) — координаты  $i$ -й точки.

Далее дано еще  $n$  строк, в  $i$ -ой из которых записано два числа  $v, u$  ( $1 \leq v, u \leq n, v \neq u$ ) — номера точек, с которыми  $i$ -я точка соединена отрезками.

Гарантируется, что точки общего положения — никакие две точки не совпадают и никакие три не лежат на одной прямой. Также все отрезки имеют ненулевую длину.

### Формат выходного файла

Если нельзя так у каждой точки удалить ровно один отрезок, в единственной строке выходного файла выведите NO. Иначе, в первой строке выходного файла выведите YES, а в  $i$ -й из следующих  $n$  строк выведите число  $x$  ( $1 \leq x \leq 2$ ) — какой из отрезков у вершины номер  $i$  надо удалить. Отрезок в вершину  $v$  имеет номер 1, отрезок в вершину  $u$  — номер 2.

### Пример

segments.in	segments.out
4	YES
0 0	1
0 1	1
1 1	2
1 0	2
2 4	
1 3	
2 4	
3 1	

## Задача Н. Маленькая шалость

Имя входного файла: `joke.in`  
Имя выходного файла: `joke.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Однажды ночью Альф решил пойти погулять. Во время прогулки пришельцу стало скучно, и он решил как-нибудь напакостить. А именно, увидев карту города, Альф решил перекрыть дорогу, причем не любую, а ту, после перекрытия которой кратчайшее расстояние от его дома до всех интересных мест города изменится у максимального числа интересных мест.

Город представлен в виде графа с  $n$  вершинами. Дом Альфа находится в вершине с номером 1.

Выведите максимальное число вершин, расстояние до которых изменится после удаления одного ребра в графе.

### Формат входного файла

В первой строке входного файла дано число  $n$  ( $1 \leq n \leq 300$ ) — количество вершин. В следующих  $n$  строках дано по  $n$  чисел  $a_{i,j}$  ( $-1 \leq a_{i,j} \leq 100000$ ) — матрица смежности. Если ребро в графе отсутствует  $a_{i,j} = -1$ . Гарантируется, что  $a_{i,i} = 0$ ,  $a_{i,j} > 0$  если  $i \neq j$ .

### Формат выходного файла

Выведите ответ на задачу.

### Пример

<code>joke.in</code>	<code>joke.out</code>
5 0 16 12 1 12 16 0 12 13 -1 12 12 0 5 2 1 13 5 0 2 12 -1 2 2 0	4

## Задача I. Сообщения

Имя входного файла: `messages.in`  
Имя выходного файла: `messages.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

На родине Альфа, планете Мелмак, для общения компьютеров по сети используется протокол МСР (Melmac Connection Protocol). Одной из подзадач при реализации протокола МСР является выбор оптимального маршрута, по которому будет передаваться сообщение.

Компьютерная сеть планеты Мелмак представляет собой связный неориентированный граф без петель и кратных ребер. Компьютер 1 хочет послать  $k$  сообщений различным компьютерам  $a_1 \cdots a_k$ . За одну секунду сообщение либо переходит по выбранному ребру, либо остается в этой вершине до следующей секунды. Два сообщения не могут в данный момент по одному ребру может проходить максимум одно сообщение. Временем передачи нескольких сообщений считается время, в которое последнее из этих сообщений достигнет адресата. Реализация протокола МСР сама может выбирать маршрут для каждого сообщения.

Вам поручено реализовать выбор оптимальных маршрутов для набора сообщений таким образом, чтобы время передачи всех сообщений было минимально.

### Формат входного файла

В первой строке входного файла заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 100$ ) — число вершин и ребер в графе соответственно.

В следующих  $m$  строках заданы пары целых чисел  $u, v$  ( $1 \leq u, v \leq n$ ) — ребра графа.

В следующей строке задано целое число  $k$  ( $1 \leq k < n$ ) — количество сообщений. В следующей строке через пробел перечислены номера вершин, в которые нужно послать сообщения  $a_i$  ( $1 < a_i \leq n$ ).

### Формат выходного файла

В единственной строке выходного файла выведите искомое время передачи сообщений.

## Пример

messages.in	messages.out
5 4 1 2 2 3 2 4 2 5 3 3 4 5	4
9 10 1 2 1 3 2 4 3 4 4 5 5 6 4 6 6 7 6 8 6 9 3 7 8 9	5

## Задача J. Диаграмма

Имя входного файла: `diagram.in`  
Имя выходного файла: `diagram.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Зайдя в спальню Вилли и Кейт, Альф случайно заметил свернутый лист бумаги. Пришелец не смог удержать желания буквально засунуть свой нос в этот сверток. Развернув его, он обнаружил, что на листе изображена какая-то диаграмма. Альф сразу же понял, что эта диаграмма принадлежит Вилли и незамедлительно решил ее исправить. Диаграмма представляет из себя набор столбцов разной высоты. Пришельцу не понравилось то, что на картинке присутствует огромное количество перепадов высоты. Он решил уменьшить это количество, уменьшив или увеличив высоты некоторых столбцов. В итоге он хочет получить диаграмму, на которой не больше  $k$  перепадов. Представим диаграмму как массив  $a_i$ , где каждый элемент обозначает высоту  $i$ -го столбца. Количество перепадов определяется как количество соседних столбцов, высоты которых различаются. Однако Альф, как заботливый пришелец, хочет сделать так, чтобы разница между исходной диаграммой и новой была минимальной. Разница между диаграммами определяется как  $\sum_{i=1}^n |a_i - b_i|$ , где  $a_i$  и  $b_i$  обозначают высоты столбцов диаграмм. Помогите Альфу найти минимальную разницу между исходной диаграммой и диаграммой, в которой не больше  $k$  перепадов высот.

### Формат входного файла

В первой строке входного файла даны два числа  $n$  и  $k$  ( $1 \leq n \leq 1000, 0 \leq k \leq 100$ ) — количество столбцов в диаграмме и максимальное количество перепадов высот. Во второй строке дано  $n$  чисел  $a_i$  ( $1 \leq a_i \leq 10^9$ ) — описание высот столбцов.

### Формат выходного файла

Выведите ответ на задачу.

### Пример

diagram.in	diagram.out
5 2 1 2 3 4 5	2
5 0 1 2 3 4 5	6