

## Задача А. Добро пожаловать на новогоднюю олимпиаду ЛКШ!

Перед тем, как перейти к решению задач, сдайте в эту задачу информацию о вашей команде.

### **Формат выходных данных**

Сдайте программу, которая выводит в произвольном формате название вашей команды, состав, а также аудиторию и номер стола, за которым вы сидите.

## Задача В. Мандарины

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Отправляясь в ЛКШ.Зима, Илья Викторович купил много мандаринов. Разместившись в вагоне, он пошёл в купе к ЛКШатам, взяв с собой новогоднее лакомство. За дружеской беседой он достал мандаринку, почистил её и поделился с Ириной. Ирина посчитала количество долек — оно оказалось чётное, она разделила мандаринку пополам и каждую половинку отдала другому ЛКШонку.

Когда в следующий раз Илья Викторович поделился мандаринкой с Ириной, число долек оказалось нечётное, и поэтому она, прежде чем разделить её на две части и отдать, съела одну дольку. Половинки она отдала тем же ЛКШатам, что и в предыдущий раз.

Более того, Илья Викторович заметил, что все ЛКШата, когда получают дольки, используют простой принцип:

- Если долек нечётное число, то ЛКШонки сначала съедает одну дольку.
- Если дольки ещё остались, то теперь их чётное число и их можно разделить на две половинки.
- Каждая половинка отдаётся другу. Интересно, что двое друзей, с которыми делится ЛКШонки, всегда одни и те же. При этом никакой ЛКШонки не оставляет долек себе и не отдаёт обе половинки одному и тому же другу.
- Если ЛКШонки передают дольки одновременно несколько друзей, то он их обрабатывает по очереди: сначала одну пачку долек, потом другую и так далее.

Илья Викторович понаблюдал за ЛКШатами в купе и записал, с кем делится каждый ЛКШонки. Теперь ему интересно, сколько долек мандарина скушает Ирина, если он отдаст одну мандаринку определённому ЛКШонки.

### Формат входных данных

В первой строке входного файла записано два натуральных числа — общее количество ЛКШат в купе  $n$  ( $3 \leq n \leq 100\,000$ ) и количество запросов  $m$  ( $0 \leq m \leq 100\,000$ ). В следующих  $n$  строках для каждого ЛКШонки указаны два числа  $a_i$  и  $b_i$  ( $1 \leq a_i, b_i \leq n$ ) — друзья, с которыми делится дольками соответствующий ЛКШонки.

В последующих  $m$  строках содержатся запросы. Каждый запрос — это два целых числа:  $x$  ( $1 \leq x \leq n$ ) — номер ЛКШонки, который первым получает мандарин, — и  $y$  ( $1 \leq y \leq 10^9$ ) — количество долек долек в мандарине (да, у Ильи Викторовича волшебные новогодние мандарины, в которых бывает миллиард долек).

### Формат выходных данных

Для каждого запроса выведите одно число — количество долек, которое скушает Ирина (ЛКШонки с номером 1).

## Примеры

стандартный ввод	стандартный вывод
3 4 2 3 1 3 1 2 1 1 2 1 3 1 2 239	1 0 0 80
3 0 2 3 1 3 1 2	
3 3 2 3 1 3 1 2 3 21 2 21 1 21	6 6 9

## Задача С. Спички — детям не игрушка!

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайта

Всем известно, что спички — детям не игрушка. Однако Антона это совершенно не остановило, когда он случайно нашел сразу  $N$  спичек. И для себя он обнаружил гораздо более увлекательное занятие, чем сжигание спичек — Антон складывает из них прямоугольники.

Длины всех спичек одинаковы и равны 1 см. При построении прямоугольников Антон не ломает спички и не обязательно использует все.

Но тут его обеспокоил вопрос — поместится ли на столе, за которым он сидит, прямоугольник с наибольшей площадью, который можно сложить из имеющихся спичек. Для начала, Антон хотел бы определить самую максимально возможную площадь прямоугольника. Помогите ему с этой задачей.

### Формат входных данных

В первой строке находится единственное целое число  $N$  ( $4 \leq N \leq 1000$ ).

### Формат выходных данных

Выведите максимальную площадь прямоугольника в  $\text{см}^2$ .

### Примеры

<code>stdin</code>	<code>stdout</code>
4	1
16	16
18	20
599	22350

## Задача D. Казус с подарками

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Доставить за ночь несколько миллионов подарков — нелегкая задача, и даже Деду Морозу порой бывает сложно с ней справиться. Поэтому из года в год он пробует различные способы улучшения процесса доставки подарков. В этом году, например, он решил испытать способ своего западного товарища Санты Клауса, который просто бросает подарки в каминные трубы домов, и так экономит кучу времени.

Сказано — сделано. Однако разложив все подарки по каминным трубам, Дед Мороз заподозрил неладное. Оказалось, что в спешке он уронил некоторые подарки не в те трубы. Нужно срочно исправлять ситуацию, чтобы Новый Год не был испорчен.

В данный момент в каждой трубе есть несколько подарков. Нужно добиться того, чтобы все подарки лежали в нужных трубах. Достать подарок из трубы можно только после того, как вынуты все подарки, лежащие над ним. Каминные трубы довольно узкие, поэтому подарки нужно доставать и складывать в них по одному. Подарки, вынутые из трубы, можно складывать в бесконечно большой мешок Деда Мороза.

Естественно, Дед Мороз хочет исправить ситуацию с подарками как можно быстрее. Помогите ему сделать это. На то, чтобы положить подарок в трубу или вынуть его оттуда, уходит одна минута. Можно считать, что Дед Мороз перемещается между трубами мгновенно. На то, чтобы положить подарок в мешок или достать его оттуда, также не уходит времени.

### Формат входных данных

В первой строке записаны два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100\,000$ ) — количество труб и количество подарков соответственно.

В следующих  $N$  строках описано состояние каминных труб на данный момент в следующем формате:

Первое число  $i$ -той строки  $k_i$  означает количество подарков в  $i$ -той трубе. Затем следуют  $k_i$  чисел — номера подарков, лежащих в этой трубе по порядку начиная с самого нижнего. Сумма всех  $k_i$  равна  $M$ .

В последней строке входных данных записано  $M$  чисел  $a_i$ , где  $1 \leq a_i \leq N$  — номер трубы, в которой должен лежать  $i$ -тый подарок.

### Формат выходных данных

Выведите одно число — минимальное время, за которое можно разложить все подарки в правильные трубы.

### Примеры

<code>stdin</code>	<code>stdout</code>
2 5 3 2 5 3 2 4 1 2 1 1 1 2	8

## Задача Е. Новогодние хлопоты

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дольше всего к Новому Году готовится, конечно, Дед Мороз, ведь ему надо подарить подарки огромному количеству детей. А прежде, чем дарить подарки их нужно выбрать, упаковать и написать на каждом адрес получателя. Но даже на этом работа не заканчивается. Прежде, чем начать развозить подарки нужно, сложить их все в большой красный мешок, подготовить сани Деда Мороза, накормить оленей и позвать Снегурочку.

Эти и другие необходимые дела секретарь Деда Мороза записал и пронумеровал. Также он выписал такие пары дел, что одно из них обязательно должно быть сделано до начала работы над вторым. Например, прежде, чем складывать подарки в мешок их нужно упаковать, а никак не наоборот. Гарантируется, что из таких зависимостей между делами никогда не следует, что некоторое дело должно быть сделано до себя самого.

Теперь секретарю интересно, за какое минимальное время можно выполнить все нужные дела. Можно считать, что на выполнение каждого дела уходит ровно один час. Помощников у Деда Мороза много, поэтому они могут работать над несколькими делами параллельно.

### Формат входных данных

В первой строке записаны два числа  $N$  и  $M$  — количество предновогодних дел и количество зависимостей между делами соответственно.  $1 \leq N, M \leq 10^5$ .

Следующие  $M$  строк описывают сами зависимости. В каждой из них записана пара чисел  $a, b$  и это означает, что дело  $a$  должно быть сделано до дела  $b$ .

### Формат выходных данных

Выведите минимальное время в часах, за которое можно закончить работу над всеми предновогодними делами.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 3 1 2 2 3 4 5	3
10 0	1

## Задача F. Ночной кошмар

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Коля проснулся среди ночи в холодном поту. Ему только что приснился кошмар, в котором он убегал от большого числа. И все бы хорошо, но со временем число становилось все больше и больше. Коля успел заметить, что на месте каждой цифры возникало число, равное ее кубу. И так каждую секунду! Коля посмотрел на часы и понял, что его сон длился  $K$  секунд. Он уже успел оправиться после кошмара, и теперь ему интересно, какой длины было нападавшее на него число в момент пробуждения. Помогите ему понять это, если известно, что изначально число было равно  $N$ .



### Формат входных данных

В единственной строке входного файла записаны два числа  $N$  и  $K$ . ( $1 \leq N \leq 100, 1 \leq K \leq 35$ )

### Формат выходных данных

Выведите одно число — длину нападавшего на Колю числа в момент пробуждения.

### Примеры

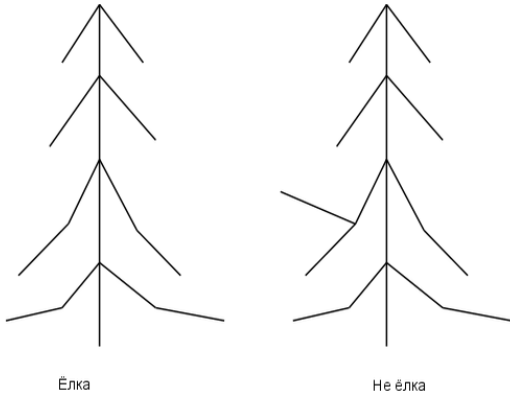
<code>stdin</code>	<code>stdout</code>
90 2	8

## Задача G. Ёлки

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Через несколько часов наступит Новый год. Люди готовятся к его встрече, режут салаты, красиво одеваются, собираются в кругу семьи и друзей. А вы почему-то все еще сидите за компьютером. Ну да не расстраивайтесь.

Вот вам новогодняя задачка — проверьте, что дерево является *елкой*. Дерево называется *елкой*, если в нем можно выделить две вершины и путь между ними (*ствол*) так, что все остальные ребра выстраиваются в цепочки, отходящие от ствола.



### Формат входных данных

В первой строке записано число  $N$  — количество вершин дерева ( $1 \leq N \leq 10\,000$ ). В следующих  $N - 1$  строках записаны пары вершин дерева, соединенных ребром.

### Формат выходных данных

Выведите «YES», если данное дерево является елкой и «NO» в противном случае.

### Примеры

stdin	stdout
2 2 1	YES
5 5 3 3 2 4 1 1 2	YES



## Задача Н. Awk

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

```
BEGIN {  
    FS="[a-zA-Z_0-9]+"  
}  
{  
    for (i = 1; i <= NF; i++)  
        if ($i ~ /^[a-zA-Z]+$/)   
            a[tolower($i)]++  
}  
END {  
    st = "sort"  
    for (w in a)  
        if (a[w] == 1)  
            print w | st;  
        else  
            print w, a[w] | st  
    close (st)  
}
```

### Примеры

стандартный ввод	стандартный вывод
kitten abc123e a_b_c, , ,eats !@mice KITten does not eat OTHER_KITTENS	does eat eats kitten 2 mice not

## Задача I. Строка

Имя входного файла: `substring.in`  
Имя выходного файла: `substring.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дед Мороз составил важное сообщение, которое нужно передать другим Дедам Морозам. На всякий случай он зашифровал его с помощью шифра замены: заменил все буквы в сообщении, одинаковые буквы на одинаковые, разные на разные. На всякий случай Дед Мороз зашифровал свое сообщение два раза.

Потом он ненадолго отвлекся, а в это время один из эльфов прибежал в дом и решил пошалить. Он взял и приписал к шифровкам спереди и сзади какую-то абракадабру.

Теперь Деду Морозу надо найти наибольшую подстроку, которая могла бы быть его шифровкой. Помогите ему.

Например, пусть сообщение Деда Мороза имело вид `'xxz'`. Он его зашифровал первый раз как `'bba'`, второй раз — как `'aab'`. После этого эльф приписал к первому сообщению спереди букву `a`, сзади — ничего; ко второму — спереди ничего, сзади `b`. В итоге получились строки `'abba'` и `'aabb'`.

### Формат входных данных

Входной файл содержит две строки, состоящие из символов с кодами от 33 до 127 и длиной не более 5000. Гарантируется, что обе строки во входном файле непустые.

### Формат выходных данных

Выведите две строки — предполагаемые шифровки, в том виде, в котором они были в первой и во второй исходных строках.

### Примеры

<code>substring.in</code>	<code>substring.out</code>
<code>abba</code> <code>aabb</code>	<code>bba</code> <code>aab</code>
<code>abab</code> <code>baba</code>	<code>abab</code> <code>baba</code>
<code>k555kk5555</code> <code>5kkk5kk5kk</code>	<code>k555k</code> <code>5kkk5</code>



## Задача К. Новогодняя сцена

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Конструируя сцену для новогоднего представления, театральный декоратору Сантону Клаусову нужно нарисовать макет сцены, на которой расположены несколько помещений, вложенных друг в друга.

Вам дана схема вложенности помещений — правильная скобочная последовательность. Выведите псевдографикой макет сцены.

### Формат входных данных

Во входном файле содержится правильная скобочная последовательность длиной не более 50 символов.

### Формат выходных данных

Выведите макет сцены с помощью символов подчеркивания («\_» — символ с кодом 95), вертикальной черты («|» — символ с кодом 124), слеша («/» — символ с кодом 47) и пробела (« » — символ с кодом 32).

Помещения «внешнего» уровня должны быть выровнены по «левой» стене. Все помещения должны быть минимально возможного размера и сдвинуты как можно сильнее «вверх» и «влево». Помещения, вложенные в одно и то же помещение, должны быть выровнены по «левой» стене вдоль «левой» стены внешнего помещения и расположены «сверху вниз» в том порядке, в котором они описаны во входном файле.

Полностью придерживайтесь формата примера выходного файла, приведенного ниже. Ответ вашей программы может отличаться от ответа жюри только наличием/отсутствием пробелов в конце строк.

### Примеры

стандартный ввод	стандартный вывод
<pre>((()())())</pre>	<pre>----- / ----- / //  _  // // / _// // /_// //  __ // //  _  // // / ----- // // /  _  // // / / _// // / /_// // /  __ // // /_// //  ----- // /_//  ----- /  --- / _// /_//  __ /</pre>

## Задача L. Билеты на сайте РЖД

Имя входного файла:	<code>stdin</code>
Имя выходного файла:	<code>stdout</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайта

Покупая через интернет билеты на поезд в ЛКШ.Зима, Михаил Эдуардович столкнулся со следующим спецэффектом: сайт `ticket.rzd.ru` «помнит» предыдущий сделанный заказ — фамилию, имя и отчество человека, для которого был куплен предыдущий билет.

М. Э. собирается купить  $4n$  билетов, сделав  $n$  заказов на четыре человека (делать заказ большего размера система не позволяет). При этом, делая каждый очередной заказ (кроме первого) при открытии формы заказа все поля оказываются автоматически заполненными данными из предыдущего заказа.

Вводя данные об очередном человеке, М. Э. копирует из текстового файла фамилию в буфер обмена и вставляет ее в форму на сайте. Это занимает у него полторы секунды. Но если фамилия в точности совпадает с той, что «помнит» сайт РЖД, то М. Э. не меняет ее, и на это у него уходит ноль секунд. Затем он аналогично поступает с именем, а затем с отчеством.

В каком порядке ему следует покупать билеты в ЛКШ.Зима, чтобы затраченное время было минимальным?

### Формат входных данных

В первой строке записано число  $n$  ( $1 \leq n \leq 4$ ) — число человек, едущих в ЛКШ.Зима, деленное на четыре. В следующих  $4n$  строчках записаны фамилии, имена и отчества, разделенные пробелами. Все фамилии, имена и отчества состоят только из латинских букв в количестве от 1 до 20, причем заглавной является только первая.

### Формат выходных данных

Выведите список людей, переупорядоченный таким образом, что покупая билеты в соответствии с этим списком (сначала первым четверым людям в указанном порядке, затем следующим четверым и т. д.), время покупки билетов станет минимальным.

Если оптимальных списков несколько, выведите любой из них.

### Примеры

<code>stdin</code>	<code>stdout</code>
2	Yuschenko Pavel Aleksandrovich
Stankevich Andrey Sergeevich	Dvorkin Mikhail Eduardovich
Sablina Margarita Markovna	Melnikov Sergey Vyacheslavovich
Kuniavskiy Pavel Evgenievich	Sayutin Dmitriy Sergeevich
Yuschenko Pavel Aleksandrovich	Kuniavskiy Pavel Evgenievich
Dvorkin Mikhail Eduardovich	Sablina Margarita Markovna
Dvorkina Tatiana Evgenievna	Dvorkina Tatiana Evgenievna
Melnikov Sergey Vyacheslavovich	Stankevich Andrey Sergeevich
Sayutin Dmitriy Sergeevich	

## Задача М. Олени Санта-Клауса

Имя входного файла: lake.in  
Имя выходного файла: lake.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Санта-Клаус едет в санях, запряженных двумя оленями. Когда он проезжал мимо места проведения ЛКШ.Зима, олени поссорились и решили дальше бежать по двум разным дорожкам. К счастью, сани у Санта-Клауса не простые, а волшебные, поэтому никакой аварии не случилось: сани просто всегда находятся точно в середине отрезка между оленями, независимо от того, куда бегут эти олени.

Санта Клаус всегда отслеживает расстояние, которое он проехал, с помощью GPS-приемника, но на этот раз оказалось, что он забыл свой приемник дома. Зато оба его оленя оснащены системой ГЛОНАСС, которая позволяет восстановить их траекторию (оказалось, что оба оленя бежали вдоль ломаных). Кроме того, известно, что олени бегут с постоянной скоростью (каждый со своей). Если олень прибегает к концу своей дорожки раньше другого оленя, он просто стоит в конце дорожки.

Помогите Санта Клаусу узнать, какое расстояние он проехал.

### Формат входных данных

В первой строке входного файла находится одно  $n$  — число вершин в ломаной, по которой бежал первый олень. Далее следуют  $n$  строк, в каждой из которых два целых числа — координаты вершин этой ломаной. Далее следует еще одна строка с числом  $m$  — числом вершин в ломаной, по которой бежал второй олень, далее следуют  $m$  строк с координатами вершин этой ломаной. В последней строке через пробел записаны скорости первого и второго оленя соответственно.

Количества вершин в каждой ломаной не превосходят 50 000, координаты не превосходят по модулю 150, скорости положительны и не превосходят 100. Все числа во входном файле целые.

Считайте, что все перемещения Санта-Клауса и оленей происходят на плоскости.

### Формат выходных данных

Выведите длину пути, пройденного санями Санта Клауса, с абсолютной или относительной точностью  $10^{-4}$ .

### Примеры

lake.in	lake.out
2 0 0 1 0 2 0 0 1 0 2 1	1.0000000000
2 0 0 0 1 2 1 0 1 1 1 1	1.0000000000

## Задача N. Магистр Йода

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Под Новый Год британские ученые раскрыли тайну речи магистра Йоды. Оказалось, что все предложения он представляет себе в виде бинарного дерева. Вершиной бинарного дерева является *самое важное слово* в предложении. Часть предложения, стоящая слева от этого слова, записывается по тем же правилам в левом поддереве, а часть предложения, стоящая правее этого слова, — в правом.

*Самое важное слово* определяется так: если в предложении есть слово `Jedi` (именно с первой заглавной, остальными маленькими буквами), то самым важным считается оно (если это слово встречается несколько раз, то самым важным считается первое из них). Для остальных слов важность слова определяется его длиной: чем длиннее слово, тем оно важнее. Из всех слов одинаковой длины самым важным считается то, которое встречается раньше.

Представив предложение в виде дерева, магистр Йода произносит его по следующему рекурсивному алгоритму: сначала он произносит правое поддерево, затем корень, а затем левое поддерево.

### Формат входных данных

Во входном файле приведено одно предложение. В предложении не более 1000 слов, в каждом слове не более 100 букв. Слова состоят из маленьких и больших латинских букв и разделяются одним пробелом; никаких знаков препинания нет.

### Формат выходных данных

Вы должны вывести, как данное предложение прочитает мастер Йода.

### Примеры

<code>stdin</code>	<code>stdout</code>
Use force Jedi	Jedi force Use
Jedi Jedi Jedi	Jedi Jedi Jedi

## Задача О. Дактерии

Имя входного файла: `matrix.in`  
Имя выходного файла: `matrix.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

На последнем снимке космического телескопа Хаббл ученые обнаружили у около одной из небольших звезд в туманности Андромеды удивительную планету, имеющую форму тора. Эта планета оказалась обитаемой. Местные жители — они называют себя донутяне — поделили всю планету на четырехугольные участки.

Проще всего представить себе это деление так: возьмем квадрат на плоскости и разделим его на  $N^2$  единичных квадратов. Теперь каждому участку соответствуют координаты  $x$  и  $y$ . Соседними являются те участки, у которых одна из координат совпадает, а другая отличается на 1, либо участки, у которых одна координата совпадает, а вторая отличается на  $N - 1$  (то есть участки, находящиеся на противоположных сторонах квадрата). Таким образом, у каждого донутянина ровно четверо соседей.

Все донутяне разводят бактерий. Год назад на Донут завезли новую разновидность бактерий — дактерии, — и каждый донутянин приобрел несколько (возможно, ноль) бактерий этого вида. За год (а год на Донуте очень короткий) число дактерий у каждого донутянина увеличилось в девять раз. И теперь, под новый год, все собрались делать друг другу подарки. Каждый Донутянин подарил всем своим четырем соседям по одной девятой от общего количества своих дактерий (оставив себе пять девятых).

Но после новогодней вечеринки все Донутяне рассорились друг с другом и решили потребовать свои подарки назад (на Земле такое представить, конечно, невозможно). Помогите им. Зная, сколько у кого оказалось бактерий после нового года, восстановите, сколько у кого из них было бактерий изначально (до того, как дактерии стали удевятеряться).

### Формат входных данных

В первой строке входного файла находится одно число  $N$  — количество участков вдоль каждой стороны квадрата. Далее следуют  $N$  строк по  $N$  чисел, указывающих количество дактерий у каждого донутянина после нового года. Гарантируется, что количество дактерий у каждого донутянина целое и не превышало  $10^9$  ни до, ни после нового года. Также гарантируется, что изначально каждый донутянин купил целое число бактерий. Гарантируется, что  $1 \leq N \leq 500$ .

### Формат выходных данных

В выходной файл выведите  $N$  строк по  $N$  чисел в каждой: количество дактерий у каждого донутянина изначально.

### Примеры

<code>matrix.in</code>	<code>matrix.out</code>
1 0	0
2 6 10 5 6	0 2 1 0
3 23 15 8 22 8 10 25 17 16	3 2 0 3 0 1 3 2 2



## Задача Р. Полотенца

Имя входного файла:	towels.in
Имя выходного файла:	towels.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

... Теперь вытереться совсем невозможно, хотя у вас и много полотенец, а все потому, что в некотором смысле между вашей собственной влажностью и влажностью полотенец нет никакой разницы. Можно придумать какую-нибудь величину, которую мы назовем «водоудаляющей способностью». Так вот, «водоудаляющая способность» полотенца равна вашей собственной «водоудаляющей способности», и, вытираясь мокрым полотенцем, вы забираете им такое же количество воды, какое забирает ваша кожа из полотенца.

...

В этом примере вода, как энергия, ведь общее количество воды не меняется. ... Так вот, если представить себе изолированную часть Вселенной и подождать достаточно долго, то из-за происходящих в этом мире случайностей энергия, как и вода, распределится по всей этой части равномерно, и от необратимости явлений не останется и следа. В этом мире никогда уже не произойдет ничего интересного в том смысле, как мы это понимаем.

---

Р. Фейнман. Характер физических законов.

После прогулки на Финский залив ЛКШонок вернулся домой насквозь мокрым: на нем было некоторое количество воды. К счастью, друзья встретили его большим запасом полотенец. К сожалению, некоторые из них уже были мокрыми: на них тоже уже содержалось некоторое количество воды.

Будем считать, что когда ЛКШонок вытирается полотенцем, суммарная масса воды, бывшей на нем и на полотенце, распределяется между ними поровну: половина воды остается на ЛКШонке, половина — на полотенце. Помогите ЛКШонку вытереться как можно суше: придумайте способ вытирания, позволяющий перенести на полотенца как можно больше воды.

### Формат входных данных

В первой строке входного файла находятся два числа  $N$  и  $M$  — число полотенец и масса воды, изначально находящейся на ЛКШонке. На второй строке входного файла находятся  $N$  чисел,  $i$ -е из которых задает массу воды, изначально находящейся на  $i$ -м полотенце. Все числа во входном файле целые, неотрицательные и не превышают 100 000.

### Формат выходных данных

В первую строку выходного файла выведите одно число  $K$  — количество операций, которые надо выполнить. Далее выведите  $K$  чисел — номера полотенец, в том порядке, как их надо использовать. Если оптимальных решений несколько, выведите любое.

Количество действий в вашем решении не должно превосходить  $5N$ . Можно доказать, что среди

оптимальных решений такое решение существует.

## Примеры

towels.in	towels.out
2 5	3
2 3	2 1 1

## Задача Q. Перекладывание подарков

Имя входного файла: `heaps.in`  
Имя выходного файла: `heaps.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

До Нового Года совсем немного времени, и Дед Мороз собирается совершить еще одну поездку. Идут последние приготовления в дорогу — скоро в сани надо складывать мешки с подарками. В наличии у Деда Мороза есть  $n$  мешков — они пронумерованы числами от 1 до  $n$ . В  $i$ -м мешке находится  $a_i$  подарков. Все подарки одинаковые. Чтобы погрузить их в сани, требуется переложить некоторые подарки между мешками так, чтобы для каждого  $i$  в  $i$ -м мешке стало  $b_i$  подарков. Деда Мороз попросил Снегурочку помочь с раскладкой подарков.

Снегурочка за одно действие может переложить ровно один подарок из любого мешка в любой другой. Но она очень суеверная, и поэтому не каждое такое перекладывание считает «правильным» и допустимым. Согласно её представлениям, есть «магическое» число  $m$ , и перекладывание невозможно, если для какого-то числа  $k$  от 0 до  $m - 1$  изменятся число мешков, количество подарков в которых при делении на  $m$  дает число  $k$ . Например, если  $a_1 = 10$ ,  $a_2 = 3$ ,  $a_3 = 2$ ,  $a_4 = 4$ ,  $a_5 = 6$  и  $m = 4$ , то перекладывание из мешка 1 в мешок 3 невозможно, так как, например, для  $k = 2$  число мешков, количество подарков в которых при делении на  $m = 4$  дает остаток 2, изменяется — с 3 до 1. В этом же примере перекладывание  $4 \rightarrow 2$  допустимо.

Теперь Снегурочку интересует вопрос, сможет ли она с помощью допустимых действий получить из исходной раскладки подарков требуемую.

### Формат входных данных

В первой строке находятся два целых числа  $n$  ( $1 \leq n \leq 10^5$ ) и  $m$  ( $1 \leq m \leq 10^{12}$ ). Во второй строке задаётся исходная раскладка —  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^{12}$ ). В третьей строке задаётся требуемая раскладка —  $n$  целых чисел  $b_1, b_2, \dots, b_n$  ( $0 \leq b_i \leq 10^{12}$ ).

### Формат выходных данных

Если переложить подарки возможно, то выведите одно слово 'YES', иначе — одно слово 'NO'.

### Примеры

<code>heaps.in</code>	<code>heaps.out</code>
3 5 1 2 3 2 3 1	YES
3 5 1 2 3 1 1 4	NO

## Задача R. Нолс

Имя входного файла: `nols.in`  
Имя выходного файла: `nols.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Простой снеговик Нолс любит все, что такое же простое, как он, и такое же круглое, как он. Он попросил своего друга, зайца Тимофея, загадать два числа,  $a$  и  $b$ . После этого он решил перемножить все *простые* числа, лежащие между  $a$  и  $b$  (включительно), и посмотреть, насколько *круглое* число получится.

К сожалению, ни Нолс, ни Тимофей сами не умеют считать. Помогите им, пожалуйста.

### Формат входных данных

В одной строке входного файла заданы два числа  $a$  и  $b$ , загаданные Тимофеем. Гарантируется, что  $1 \leq a \leq b \leq 10^9$ .

### Формат выходных данных

Выведите количество нулей, которые будут на конце произведения всех простых чисел, лежащих между  $a$  и  $b$  включительно.

### Примеры

<code>nols.in</code>	<code>nols.out</code>
1 5	1

## Задача S. Снежинка

Имя входного файла: `snowflake.in`  
Имя выходного файла: `snowflake.out`  
Ограничение по времени: 10 секунды  
Ограничение по памяти: 64 мегабайта

Мальчик N. поехал в ЛКШ.Зима, но, к сожалению, его девушку, M., на школу не взяли. И теперь все время, когда N. не пишет код, он пишет письма M., в которых подробно описывает все, что происходит на школе. Особенно подробно он описывает снежинки (поскольку их здесь не слишком много). Для описания снежинок он выбрал следующий формат.

Снежинка, как известно, состоит из шести одинаковых лучей, симметрично выходящих из центра. Будем считать, что каждый луч снежинки имеет следующую структуру. Он либо пустой (то есть не содержит ни одной молекулы воды), либо имеет следующий вид: ледяная палочка некоторой длины вдоль направления луча, потом разветвление. Из разветвления торчат пять лучей: четыре под углами  $60^\circ$ ,  $-60^\circ$ ,  $120^\circ$ ,  $-120^\circ$  к направлению луча, и один дальше по направлению луча. Будем для простоты считать, что лучи на  $60^\circ$  и  $-60^\circ$  одинаковы, равно как и лучи на  $120^\circ$  и  $-120^\circ$ . Тогда луч можно задавать строкой следующим образом: луч это либо пустая строка, либо последовательность вида

`число (луч) (луч) луч`

Число обозначает длину отрезка до разветвления. В скобках сначала записывается луч, идущий под углом  $120^\circ$  (то есть вбок и назад), потом луч, идущий под углом  $60^\circ$  (т.е. вбок и вперед). После скобок — продолжение луча прямо.

M. любит снег крупными хлопьями и хочет определить по описанию N., насколько большие снежинки встречаются в Лисьем Носу. Напишите программу, которая по данному описанию снежинки определяет расстояние от ее центра до самой удаленной от центра точки.

### Формат входных данных

Во входном файле находится одна строка — описание снежинки в указанном формате. Длина строки не превосходит 100, длина каждого ребра не превосходит 1000. Строка не является пустой и не содержит никаких символов, кроме цифр и круглых скобок.

Отдельные части луча снежинки могут пересекаться, равно как и отдельные части разных лучей снежинки (напомним, что целиком снежинка состоит из шести идентичных лучей, выходящих из центра). Вы можете игнорировать подобные пересечения.

### Формат выходных данных

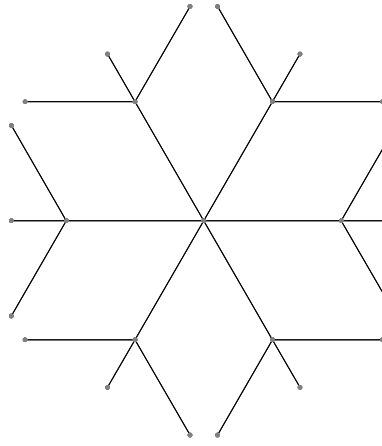
Выведите в выходной файл одно число — радиус снежинки с абсолютной или относительной погрешностью не более  $10^{-6}$ .

### Примеры

<code>snowflake.in</code>	<code>snowflake.out</code>
<code>5() (4() ()) 2() ()</code>	7.8102496759
<code>5(4() ()) (4() ()) 2() ()</code>	7.8102496759

### Замечание

Снежинка из первого примера



## Задача Т. Многоуровневая Математика Манипуляций

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В Летней Компьютерной Школе была запущена реферральная система для привлечения новых школьников: в конце смены к количеству решённых вами *задач* будет добавлено  $p_1\%$  от количества задач всех приглашённых вами ЛКШат. Более того, если какой-то приглашённый вами школьник сам приведёт новых ЛКШат (назовём их вашими друзьями второго уровня), то от каждого из них вы получите  $p_2\%$  решённых ими задач. А если и они приведут новых друзей (третьего уровня), то вы получите  $p_3\%$  их задач. Система работает аналогично до друзей  $k$ -го уровня включительно.

Более формально, назовём друзьями первого уровня ЛКШат, которые были приглашены в ЛКШ непосредственно вами, а друзьями  $i$ -го ( $i \geq 2$ ) уровня — ЛКШат, которые были приглашены в ЛКШ вашими друзьями уровня  $i-1$ . За каждого друга  $i$ -го уровня количество решённых вами задач будет увеличено в конце смены на  $p_i\%$  от решённых им задач для всех  $1 \leq i \leq k$ . За друзей уровня  $k+1$  и последующих никаких бонусов вам не полагается.

У вас есть  $n$  знакомых, которые ещё не были в ЛКШ. Про каждого из них вы заранее знаете, сколько задач он решит за смену. Вы уже были в ЛКШ, поэтому ваши знакомые безропотно вам подчиняются и вы можете указать им, как именно они должны раздать друг другу приглашения в ЛКШ. Требуется определить способ раздачи приглашений, который обеспечит вам максимальное увеличение числа решённых вами задач в конце смены.

Обратите внимание, что благодаря данной реферральной системе количество задач, решённых вами в конце смены, не обязательно будет целым.

### Формат входных данных

В первой строке заданы два числа  $n$  и  $k$  ( $1 \leq n, k \leq 1000$ ) — количество ваших знакомых и максимальный уровень друга, влияющего на ваш результат в конце смены.

Во второй строке задано  $n$  чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 1000$ ) — число задач, которые решит каждый из ваших друзей.

В третьей строке задано  $k$  чисел  $p_1, p_2, \dots, p_k$  ( $0 \leq p_i \leq 100$ ) — количество начисляемых процентов с решённых задач друзьями первого, второго и так далее до  $k$ -го уровня соответственно.

### Формат выходных данных

Выведите  $n$  чисел, по одному в строке, определяющих оптимальный набор приглашений между вашими друзьями.  $i$ -е число должно быть равно 0, если  $i$ -го знакомого должны пригласить непосредственно вы, либо номеру знакомого, который должен его пригласить. Обратите внимание, набор приглашений должен быть устроен таким образом, чтобы каждый школьник оказался вашим другом некоторого порядка.

Если оптимальных ответов несколько, разрешается вывести любой из них.

## Примеры

стандартный ввод	стандартный вывод
6 4 10 20 30 40 50 70 15 0 70 60	2 0 1 1 1 1
2 3 4 5 6 7 8	0 1
3 3 3 4 5 3 1 4	0 0 0



## Задача U. Цепочка преподавателей

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

Вам нужно составить цепочку длины хотя бы 8 из имён и фамилий преподавателей этой смены. Все использованные вами строчки должны быть попарно различны. Последняя буква каждого слова должна совпадать с первой буквой следующего слова. Например:

Демид  
Денис  
Саютин

### Формат выходных данных

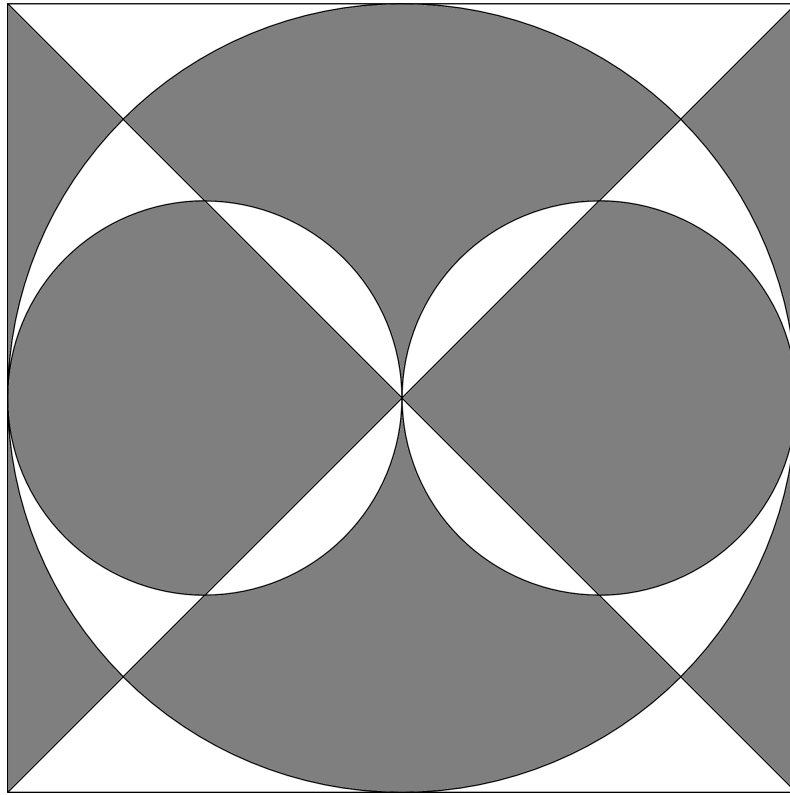
Сдайте программу, которая на каждой строке выходного файла выводит имя или фамилию очередного преподавателя в цепочке.

### Замечание

Сохраняйте файл с решением в кодировке UTF-8.

## Задача V. Г — геометрия

Найдите отношение суммарной площади белых областей к суммарной площади серых областей.



### Формат выходных данных

Сдайте программу, которая выводит одно вещественное число. Относительная погрешность ответа не должна превышать  $10^{-4}$ .



