

## Задача А. Эпическая потасовка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Легендарные герои Бандергольф Сайберсвитч и Бенадрил Кэббэджджэтч, не сумев поделить между собой роль детектива в известном кино, положились на старый метод решения споров — обнажили свои зачарованные мечи и пустились сражаться, нанося удар за ударом.

Состояние героя-воина можно задать целым числом  $h$  — его *здоровьем*, а силу чар, наложенных на оружие, натуральным числом  $a$  — его *атакой*. Изначально здоровье является натуральным числом. Когда герой получает удар оружием, его здоровье уменьшается на величину, равную атаке оружия. Если после этого здоровье перестаёт быть положительным, то герой больше не может атаковать и выходит из боя, а его противник объявляется победителем.

Герои наносят друг другу удары по очереди, первую атаку совершил Бандергольф. Помогите многочисленным наблюдателям предугадать исход этой эпической потасовки — выясните, кто выйдет из неё победителем и получит роль в кино.

### Формат входных данных

Первая строка входных данных содержит натуральные числа  $h_1$  и  $a_1$ , разделённые пробелом ( $1 \leq h_1, a_1 \leq 2 \cdot 10^9$ ) — здоровье Бандергольфа и атаку его меча. Вторая строка в том же формате содержит здоровье  $h_2$  и атаку  $a_2$  меча Бенадрила.

### Формат выходных данных

Выведите «Bandergolf Cyberswitch», если Бандергольф Сайберсвитч победит в потасовке. В противном случае выведите «Benadryl Cabbagepatch».

### Система оценки

Задача содержит две подзадачи. Для того, чтобы решение было принято на проверку, необходимо, чтобы оно проходило **все** тесты из условия.

#### Подзадача 1 (65 баллов)

Для всех тестов данной подзадачи  $1 \leq h_1, a_1, h_2, a_2 \leq 100$ . Баллы за эту подзадачу будут начислены в случае прохождения всех тестов из условия и всех тестов этой подзадачи.

#### Подзадача 2 (35 баллов)

Дополнительных ограничений нет. Баллы за эту подзадачу будут начислены в случае прохождения всех тестов из условия, всех тестов предыдущей подзадачи и всех тестов этой подзадачи.

### Примеры

стандартный ввод	стандартный вывод
7 10 7 2	Bandergolf Cyberswitch
2 9 10 2	Benadryl Cabbagepatch
5 1 5 1	Bandergolf Cyberswitch

### Замечание

В первом тесте из примера атака меча Бандергольфа, равная 10, превосходит здоровье Бенадрила, равное 7, поэтому после первого же удара Бенадрил выйдет из боя.

Во втором тесте Бенадрил выдерживает первый удар Бандергольфа (и его здоровье становится равным  $10 - 9 = 1$ ), поэтому он атакует Бандергольфа. Так как после этого здоровье Бандергольфа становится нулевым, Бенадрил побеждает.

В третьем тесте после четырёх пар ударов здоровья обоих соперников становятся равными 1, и после пятой атаки Бандергольфа Бенадрил выбывает.

## Задача В. Весы и монеты

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

У Коли есть мешок с  $n$  монетами. Все монеты неотличимы на вид, но одна из них фальшивая. Известно, что фальшивая монета либо легче, либо тяжелее настоящей.

У Коли под рукой оказался прототип нового изобретения — весы с  $k$  чашами. Если положить любое количество монет (возможно, ни одной) на каждую чашу и нажать на кнопку, то произойдет взвешивание: чаша с более лёгким грузом окажется выше чаши с более тяжёлым, а чаши с одинаковым грузом расположатся на одном уровне. Этот прототип работает очень медленно: на каждое взвешивание требуется целая минута.

Помогите Коле узнать минимальное время, которого гарантированно хватит, чтобы найти фальшивую монету.

### Формат входных данных

В одной строке через пробел подаются целые числа  $k$  и  $n$  — количество чаш у весов и количество монет у Коли, соответственно.

### Формат выходных данных

Выведите одно число — необходимое количество минут.

### Система оценки

Задача содержит четыре подзадачи. Баллы за каждую подзадачу будут начислены в случае прохождения всех тестов соответствующей подзадачи.

#### Подзадача 1 (20 баллов)

Для всех тестов данной подзадачи  $k = 3$ ,  $4 \leq n \leq 4^{14}$ ,  $n$  — степень четверки.

#### Подзадача 2 (20 баллов)

Для всех тестов данной подзадачи  $3 \leq k \leq 10^9$ ,  $4 \leq n \leq 10^9$ ,  $n$  — степень числа  $k + 1$ .

#### Подзадача 3 (30 баллов)

Для всех тестов данной подзадачи  $3 \leq k \leq 10^9$ ,  $3 \leq n \leq 10^9$ .

#### Подзадача 4 (30 баллов)

Для всех тестов данной подзадачи  $3 \leq k \leq 10^{18}$ ,  $3 \leq n \leq 10^{18}$ .

### Пример

стандартный ввод	стандартный вывод
3 4	1

### Замечание

В примере мы раскладываем по одной монете на каждую чашу весов и одну монету не кладем ни на одну из чаш.

После взвешивания возможны две ситуации:

- Все чаши весов оказались на одном уровне. Тогда фальшивая монета — та, которую мы никуда не клали.
- Одна из чаш весов оказалась выше или ниже остальных. Тогда на ней и лежит фальшивая монета.

Таким образом, нам хватило одного взвешивания, чтобы найти фальшивую монету.

## Задача С. Индийская клавиатура

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Гриша недавно устроился стажером в индийскую IT-компанию Roorkee. Его обязанности как стажера невелики — требуется всего лишь оцифровывать тексты с изречениями великих индийских мудрецов древности.

Добравшись до своего рабочего места, Гриша был весьма неприятно удивлен — оказалось, что вместо привычных нам клавиатур индийские программисты используют огромную прямоугольную клавиатуру размера  $n \cdot m$ , в каждой ячейке которой содержится уникальная буква из некоторого цифрового алфавита, имеющая номер от 1 до  $n \cdot m$ .

Окинуть всю клавиатуру взглядом может быть затруднительно, поэтому в любой момент времени Грише доступен лишь некоторый подпрямоугольник размера  $p \cdot q$  (который он волен выбрать по своему желанию). Разумеется, при печати Гриша может использовать только те буквы, которые попадают в выбранный им подпрямоугольник. Более того, за одно действие он может переконфигурировать клавиатуру: изменить подпрямоугольник, тем самым изменив набор доступных букв. Изначальную конфигурацию Гриша выбирает на свое усмотрение (но на это не тратится дополнительное действие).

Изречение, которое ему предстоит напечатать сегодня, состоит из  $s$  букв, принадлежащих тому же цифровому алфавиту, что и буквы на клавиатуре. В целях оптимизации процесса Грише хочется как можно меньше раз перенастраивать клавиатуру в процессе набора текста. Помогите ему посчитать, сколько раз ему таки придется это делать, если действовать оптимально.

### Формат входных данных

В первой строке заданы четыре числа  $n, m, p, q$  ( $1 \leq p \leq n \leq 500, 1 \leq q \leq m \leq 500$ ) — размеры клавиатуры и размеры участка клавиатуры, который видит Гриша.

В следующих  $n$  строках задано по  $m$  чисел — буквы в соответствующих ячейках клавиатуры.

Далее следует число  $s$  ( $1 \leq s \leq 3 \cdot 10^5$ ) — размер текста, который предстоит набрать Грише.

В последней строке через пробел следуют  $s$  букв  $a_i$  — сам текст. Гарантируется, что  $1 \leq a_i \leq n \cdot m$ .

### Формат выходных данных

Выведите одно число — минимальное количество переконфигураций клавиатуры.

### Система оценки

Задача содержит три подзадачи.

#### Подзадача 1 (20 баллов)

Для всех тестов данной подзадачи  $1 \leq n, m \leq 20, 1 \leq s \leq 100$ .

Данная подзадача является обязательной — без прохождения решением **всех** тестов первой подзадачи последующие подзадачи не засчитываются.

#### Подзадача 2 (30 баллов)

Для всех тестов данной подзадачи  $1 \leq n, m \leq 100, 1 \leq s \leq 1000$ .

#### Подзадача 3 (50 баллов)

Дополнительных ограничений нет.

Баллы за последнюю подзадачу будут начислены только в случае прохождения **всех** тестов этой подзадачи и **всех** тестов предыдущих подзадач.

## Пример

стандартный ввод	стандартный вывод
2 2 1 1 1 3 2 4 5 1 1 3 2 3	3

## Замечание

В первом примере в качестве начальной конфигурации Гриша волен взять подпрямоугольник размера  $1 \cdot 1$ , содержащий букву 1.

Тогда первые две буквы он сможет напечатать без перенастройки клавиатуры, а затем ему придется трижды последовательно перенастраивать клавиатуру на подпрямоугольники с буквами 3, 2 и снова 3.

## Задача D. Киви

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

Впервые попав на экскурсию по плантациям южного Китая, Олег Александрович был ошарашен — он никогда не думал, что на самом деле киви растут на деревьях! Формально, дерево киви представляет собой дерево из  $n$  вершин с корнем в вершине 1, в каждой вершине которого находится по сладкому плоду.

Видя искренний интерес Олега Александровича, работники плантации предложили ему помочь в обработке киви. Более точно, в процессе обработки в дереве киви выделяется некоторое связное поддерево, а остальная часть отбрасывается за ненадобностью.

Основываясь на рыночных данных, лучше всего продаются обработанные деревья, представляющие собой полные  $k$ -ичные кучи для некоторых  $k$ . Формально, дерево с корнем в вершине  $x$  является  $k$ -ичной кучей глубины  $m$ , если оно удовлетворяет следующим условиям:

- При  $m = 1$   $k$ -ичной кучей глубины 1 является сама вершина  $x$ .
- При  $m > 1$  дерево с корнем в  $x$  является  $k$ -ичной кучей глубины  $m$ , если у вершины  $x$  существует хотя бы  $k$  потомков  $v_i$ , так что поддерево с корнем в вершине  $v_i$  является  $k$ -ичной кучей глубины  $m - 1$ .

Сотрудники плантации просят Олега Александровича помочь им и для каждого  $k$  от 1 до  $n$  определить, как необходимо обработать изначальное дерево, чтобы получившееся поддерево образовывало  $k$ -ичную кучу наибольшей возможной глубины.

Обратите внимание, что никаких изменений к изначальному дереву не применяется, то есть для всех  $k$  задача решается независимо.

### Формат входных данных

В первой строке задано число  $n$  ( $1 \leq n \leq 10^5$ ) — размер дерева.

В следующих  $n - 1$  строках задано по два числа  $u$  и  $v$  ( $1 \leq u, v \leq n$ ), описывающих вершины, соединенные ребрами. Гарантируется, что заданный набор ребер образует дерево.

### Формат выходных данных

Выведите  $n$  чисел через пробел, где  $i$ -е означает максимальную глубину  $i$ -ичной кучи, которую можно получить из заданного дерева.

### Система оценки

Задача содержит четыре подзадачи.

#### Подзадача 1 (10 баллов)

Для всех тестов данной подзадачи  $1 \leq n \leq 5$ .

#### Подзадача 2 (10 баллов)

Для всех тестов данной подзадачи  $1 \leq n \leq 10^5$ , при этом  $i$ -е ребро (в 1-индексации) соединяет вершины  $i$  и  $i + 1$ .

#### Подзадача 3 (50 баллов)

Для всех тестов данной подзадачи  $1 \leq n \leq 2500$ .

#### Подзадача 4 (30 баллов)

Дополнительных ограничений нет.

Баллы за последнюю подзадачу будут начислены только в случае прохождения **всех** тестов этой подзадачи и **всех** тестов предыдущих подзадач.

## Примеры

стандартный ввод	стандартный вывод
2 1 2	2 1
5 1 2 2 3 2 4 1 5	3 2 1 1 1

## Замечание

В первом примере 1-ичную кучу можно получить в вершине 1. Для 2-ичной кучи все варианты равнозначны.

Во втором примере 1-ичную кучу опять же можно получить в вершине 1. Для 2-ичной кучи разумным выбором будут вершины 1 или 2; для получения остальных куч дерево можно обрабатывать произвольным образом.