

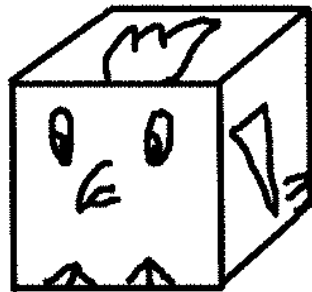
## Задача А. Бассейн для питона

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

Питон по имени Левиафан очень любит плавать. Чтобы в любой момент иметь возможность заняться любимым делом, он решил построить себе бассейн. Левиафану обычно нравится плавать дистанции длиной в  $s$  попугаев, причём во всех направлениях. Как известно, длина питона составляет 38 попугаев, поэтому он решил сделать бассейн в форме куба со стороной  $38 + s$  попугаев.

Так как Левиафан не хочет самостоятельно рыть яму для бассейна, то он решил обратиться за помощью в компанию «Полосатый слон». За каждый кубический метр объёма бассейна питону придётся отдать 7 бананов.

Хотя Левиафану известно, что в одном кубическом метре 125 кубических попугаев, его математических способностей не хватает для вычисления количества бананов, которые нужно отдать «Полосатому слону». Помогите ему!



Кубический попугай

### Формат входных данных

В единственной строке задано число — любимая Левиафаном длина дистанции в попугаях.

### Формат выходных данных

Выведите одно число — количество бананов, которое питону придётся отдать компании «Полосатый слон» ( $\frac{(s+38)^3}{125} \cdot 7$ ).

### Примеры

<code>stdin</code>	<code>stdout</code>
12	7000.0
42.5	29212.967

## Задача В. Обед для питона

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

Питон Монти привык каждый день кушать кроликов. Для этого он заготовил  $A$  кроликов. Известно, что хрупкий организм питона требует потребления  $B$  кроликов ежедневно. И вот Монти задумался: на сколько дней ему хватит этого запаса. Также Монти хочется знать, сколько кроликов ему достанется на самый последний обед.

### Формат входных данных

На вход подаются через пробел два числа:  $A$  и  $B$  ( $1 \leq A, B \leq 10^{1000}$ )

### Формат выходных данных

Выведите два числа — результат целочисленного деления  $A$  на  $B$  и остаток от этого деления.

### Примеры

<code>stdin</code>	<code>stdout</code>
5 2	2 1
9 3	3 0

## Задача С. Лекция питона

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

Мудрый питон Мухамедгариф частенько читает лекции школьникам. Но когда его ученики записывают то, что он говорит, получается не совсем то, что он хотел сказать.

Питон любит удлинять шипящие буквы. Если слово заканчивается на «s», то он повторяет две последние буквы слова ещё раз.

Питон считает, что память у учеников короткая. Если длина произносимого слова оказывается строго больше десяти, то питон повторяет слово целиком.

Вот и сегодня, когда питон посмотрел на конспекты школьников, пытаясь вспомнить, на чём он остановился вчера, он не смог ничего понять. Что же он увидел в тетрадях школьников?

### Формат входных данных

В единственной строке дано одно из слов ненулевой длины, которое питон сказал на прошлой лекции.

### Формат выходных данных

Выведите содержимое конспектов учеников.

### Примеры

<code>stdin</code>	<code>stdout</code>
<code>hello</code>	<code>hello</code>
<code>shoes</code>	<code>shoeses</code>
<code>pleasebequiet</code>	<code>pleasebequietpleasebequiet</code>
<code>analytics</code>	<code>analyticscsanalyticscs</code>
<code>s</code>	<code>ss</code>

### Замечание

Если слово состоит из одной буквы «s», то оно удваивается один раз.

## Задача D. Примени перестановку

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

На олимпийских соревнованиях по плаванию только что завершился очередной заплыв, и Петя записал, в каком порядке финишировали спортсмены на дорожках. Запись «2 0 1» означает, что золото выиграл спортсмен, плывший по второй дорожке, серебро — плывший по нулевой (внимание: дорожки нумеруются с нуля) и бронзу — по первой. Также у Пети есть стартовый протокол, в котором указано, кто на каких дорожках стартовал. Протокол «Phelps Thorpe Popov» соответствует следующему начальному расположению: Phelps стартует на нулевой дорожке, Thorpe — на первой, Popov — на второй.

Теперь Петя хочет составить итоговую таблицу результатов: фамилии спортсменов в том порядке, в котором они финишировали. Помогите ему.

### Формат входных данных

В первой строке входных данных заданы  $n$  ( $n \leq 100$ ) чисел  $a_0, \dots, a_{n-1}$  — порядок, в котором финишировали спортсмены на дорожках. Гарантируется, что каждое число от 0 до  $n-1$  встречается ровно один раз.

Во второй строке задан стартовый протокол. Это  $n$  фамилий (каждая из них — комбинация латинских букв и цифр без пробелов), разделенных пробелом.

### Формат выходных данных

Выведите итоговую таблицу результатов — фамилии спортсменов в том порядке, в котором они доплыли до финиша. Фамилии выводите на одной строке и разделяйте пробелом.

### Примеры

stdin	stdout
2 0 1 Phelps Thorpe Popov	Popov Phelps Thorpe
0 4 2 5 3 1 Yu Cseh Zige Yi Lochte Cseh	Yu Lochte Zige Cseh Yi Cseh

## Задача E. Реверс удавов

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

На каждом удаве из стаи написано его имя. Имя удава написано маленькими латинскими буквами от головы к хвосту. Все удавы из стаи ползут друг за другом, ведь так легче ползти. Иногда вожак стаи дает команду «Реверс». В этом случае каждый удав стаи разворачивается, и стая начинает ползти в противоположном направлении. Название стаи можно прочитать, если читать от головы удава, ползущего первым, к хвосту последнего. При этом название может измениться после команды «Реверс». Имена же удавов не меняются.

### Формат входных данных

Первая строка содержит одно число  $1 \leq N \leq 100\,000$  — количество удавов. В следующих  $N$  строках написаны имена удавов в том порядке, в котором они ползут. Имя удава — строка, содержащая не более 10 маленьких латинских букв.

### Формат выходных данных

Выведите единственную строку — название стаи после команды «Реверс».

### Примеры

<code>stdin</code>	<code>stdout</code>
3 abc def ghi	ghidefabс

## Задача F. Имя для питона

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

Питоны очень внимательно относятся к выбору своих имён. Сначала для каждого детёныша его мать придумывает последовательность маленьких латинских букв по известному только питонам правилу. Однако в результате часто получается труднопроизносимое слово. Поэтому в полученной последовательности труднопроизносимые буквы надо заменить на любимую питонами букву *A*. Но это не всегда возможно, так как древний питоний закон гласит, что нельзя менять буквы, стоящие на позициях  $C[i]$  (по древней традиции, позиции нумеруются, начиная с нуля).

### Формат входных данных

В первой строке записано слово  $S$  ( $1 \leq \text{len}(S) \leq 100$ ), придуманное матерью питона и состоящее из маленьких латинских букв. Во второй строке через пробел перечислены буквы  $B[i]$  ( $1 \leq i \leq 25$ ), труднопроизносимые для питонов. В третьей строке через пробел перечислены позиции  $C[i]$  ( $0 \leq C[i] < \text{len}(S), 0 \leq i \leq \text{len}(S) \leq 100$ ), изменения в которых запрещены древним законом. В последней строке находится любимая питонами буква  $A$  ( $A \neq B[i]$  при любом  $i$ ).

### Формат выходных данных

Выведите одну строку — имя питона после преобразований.

### Примеры

<code>stdin</code>	<code>stdout</code>
<code>james m r t e s</code>	<code>jasss</code>
<code>niishka k i 1 2 h</code>	<code>niishha</code>

## Задача G. За мной просили не занимать

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

На одну ленту, вставляемую в кассовый аппарат, помещается  $L$  строк текста. Описание каждого приобретаемого товара занимает в чеке  $m$  строк, никакой дополнительной информации в чеке не печатается. Кассир меняет ленту либо в конце дня, либо когда видит, что покупки следующего покупателя не поместятся на ленте. Ваша задача — определить, после какого покупателя кассиру придется впервые за день поменять ленту в аппарате

### Формат входных данных

В первой строке заданы два числа: длина ленты  $L$  ( $1 \leq L \leq 1000$ ) и количество строк, которые занимает один товар ( $1 \leq m \leq 50$ ).

Во второй строке задано число  $n$  — количество покупателей, пришедших в течение дня ( $1 \leq n \leq 1000$ ).

В третьей — через пробел указано количество товаров, приобретаемых каждым покупателем. Гарантируется, что покупки каждого покупателя могут уместиться на одной ленте.

### Формат выходных данных

Одно число — номер покупателя, после которого придется поменять ленту (покупатели нумеруются с 1). Гарантируется, что ленту придется поменять.

### Примеры

<code>stdin</code>	<code>stdout</code>
10 1 5 2 4 3 5 1	3
7 2 4 1 2 1 1	2

### Замечание

Во втором тесте: первые два покупателя совершили в сумме 3 покупки и заняли на ленте 6 строк. Осталась одна свободная строка. Покупки третьего покупателя занимают 2 строки и не влезают на ленту, следовательно перед ним ленту надо менять.

## Задача Н. Гарри Поттер и три заклинания

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

Давным-давно (наверное, еще в первой книге) великий алхимик, создатель философского камня Николас Фламель научил Гарри Поттера трем полезным заклинаниям. Первое из них позволяет превратить  $a$  граммов песка в  $b$  граммов свинца, второе —  $c$  граммов свинца в  $d$  граммов золота и третье —  $e$  граммов золота в  $f$  граммов песка. Когда Гарри рассказал об этих заклинаниях своим друзьям, Рон Уизли был в восторге: ведь если получится превращать песок в свинец, свинец в золото, часть золота — снова в песок и так далее, то можно будет, начиная с небольшого количества песка, получить огромное количество золота! Даже бесконечное количество золота! Гермиона Грейнджер, напротив, отнеслась к этой идее скептически. Она утверждает, что согласно закону сохранения материи невозможно получить бесконечное количество материи даже при помощи магии. Наоборот, количество материи при превращениях может даже уменьшаться, переходя в магическую энергию. Несмотря на то, что аргументы Гермионы выглядят убедительно, Рон не собирается ей верить. По его мнению, Гермиона придумала свой закон сохранения материи только для того, чтобы Гарри с Роном перестали заниматься ерундой, а лучше шли учить уроки. Поэтому Рон уже набрал некоторое количество песка для экспериментов и, кажется, ссоры между друзьями не избежать...

Помогите Гарри определить, кто из его друзей прав, и все-таки предотвратить ссору. Для этого вам придется выяснить, можно ли из некоторого конечного количества песка получить количество золота, большее любого наперед заданного числа.

### Формат входных данных

В первой строке заданы 6 целых чисел  $a, b, c, d, e, f$  ( $0 \leq a, b, c, d, e, f \leq 1000$ ).

### Формат выходных данных

Выведите «Ron», если имея некоторое конечное количество песка (и не имея вообще золота и свинца), возможно получить сколь угодно большое количество золота, т. е. прав Рон. В противном случае выведите «Hermione».

### Примеры

stdin	stdout
100 200 250 150 200 250	Ron
100 50 50 200 200 100	Hermione
100 10 200 20 300 30	Hermione
0 0 0 0 0 0	Hermione

### Замечание

Разберем первый пример. Начнем с 500 граммов песка. Применяя 5 раз первое заклинание, превратим их в 1000 граммов свинца. Затем 4 раза применим второе заклинание и получим 600 граммов золота. Из них выделим 400 и превратим их снова в песок. Получим 500 граммов песка и 200 граммов золота. Применяя все те же операции к 500 граммам песка повторно, можно будет каждый раз получать дополнительные 200 граммов золота. Таким образом можно получить 200, 400, 600 и т.д. граммов золота, т.е. начиная с конечного количества песка (500 граммов) можно получить количество золота, большее любого наперед заданного числа.

В четвертом примере, применяя заклинания, невозможно получить ни песок, ни свинец, ни золото.