

## Задача А. Волчья доля

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

Три поросёнка собирали яблоки. Первый поросёнок собрал  $a$  яблок, второй —  $b$ , третий —  $c$ . Они хотят урожай разделить поровну, не разрезая яблок. Кроме того, часть яблок (не меньше половины) придётся отдать волку за охрану сада. Какое минимальное количество яблок придётся отдать? Яблоки будем считать одинаковыми.

### Формат входных данных

Введите три целых положительных числа  $a$ ,  $b$ ,  $c$ , не превосходящих  $10^9$ . Сумма этих чисел не меньше 6.

### Формат выходных данных

Выведите искомую величину — количество яблок, доставшихся волку.

### Пример

стандартный ввод	стандартный вывод
3 5 7	9

## Задача В. Красивые числа

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

Число назовём *красивым*, если в его десятичной записи на чётных позициях стоят чётные цифры, а на нечётных – нечётные. Цифры 0, 2, 4, 6, 8 считаются чётными, остальные цифры – нечётными. Нумерация позиций цифр в числе начинается с единицы и ведётся справа налево.

Вам требуется определить, сколько из  $N$  введённых чисел являются красивыми.

### Формат входных данных

Первая строка входных данных содержит величину  $N$  ( $2 \leq N \leq 10$ ). В каждой из  $N$  последующих строк записано очередное число — целая положительная величина, не превосходящая  $10^9$ .

### Формат выходных данных

В единственной строке выведите искоемое количество красивых чисел.

### Примеры

стандартный ввод	стандартный вывод
2 8741 1513	1
4 6 7 8 9	2

### Замечание

Пояснение к первому примеру: число 8741 — красивое (1 на первой позиции, 4 на второй, 7 на третьей и 8 на четвёртой), а 1513 — нет.

## Задача С. Проблема Гольдбаха

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

Проблема Гольдбаха — одна из самых старых проблем математики, не разрешённых до сих пор. В 1742 году прусский математик Кристиан Гольдбах послал Леонарду Эйлеру письмо, в котором было высказано предположение: любое чётное число, большее двух, можно представить в виде суммы двух простых чисел.

Напомним, что простое число — это натуральное число, большее единицы, имеющее ровно два натуральных делителя: единицу и само себя.

К настоящему времени это утверждение ни доказано, ни опровергнуто, хотя на апрель 2012 года известно, что оно выполняется для всех чётных чисел, не превышающих  $4 \cdot 10^{18}$ . Вам необходимо представить заданное чётное число  $N$ , большее двух, в виде суммы двух простых чисел  $a_1$  и  $a_2$ . При этом  $a_1$  должно быть минимальным из возможных чисел, и  $a_1 \leq a_2$ .

### Формат входных данных

Введите значение числа  $N$  ( $4 \leq N \leq 10^6$ ).

### Формат выходных данных

Выведите два числа  $a_1$  и  $a_2$ , разделённые единственным пробелом.

### Примеры

стандартный ввод	стандартный вывод
4	2 2
10	3 7
32	3 29

## Задача D. «Жизнь»

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

«Жизнь» – компьютерная игра, придуманная английским математиком Джоном Конвеем (John Horton Conway). Впервые описание этой игры было опубликовано в октябрьском выпуске (1970) журнала *Scientific American*, в рубрике «Математические игры» Мартина Гарднера (Martin Gardner). Рассмотрим одну из разновидностей этой игры.

Место действия игры – «вселенная» – размеченный на единичные квадраты (клетки) прямоугольник размером  $M$  строк и  $N$  столбцов. Каждая клетка может находиться в одном из двух состояний: быть живой или мёртвой. Клетка имеет, в зависимости от ее расположения, от трех до восьми соседей (т.е. клеток, имеющих с ней общую сторону или угол). Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по следующим правилам:

- вначале наступает фаза рождения. Каждая пустая (мёртвая) клетка, рядом с которой есть ровно три живые клетки-соседки, оживает;
- затем наступает фаза смерти. Если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседей меньше двух или больше трёх) клетка умирает от «одиночества» или «перенаселённости». Клетки, ожившие на фазе рождения этого же поколения, не учитываются при расчёте количества живых соседей.

По карте вселенной с первым поколением клеток определите карту с  $P$ -м поколением.

### Формат входных данных

Первая строка входных данных содержит три целых числа – величины  $M, N, P$  ( $1 \leq M, N \leq 100, 1 \leq P \leq 1000$ ). Далее следуют  $M$  строк по  $N$  символов каждая – описание первого поколения. Символ '\*' (звёздочка) в этом описании соответствует живой, а символ '.' (точка) – мёртвой клетке.

### Формат выходных данных

Выходные данные должны содержать описание  $P$ -го поколения в том же формате, что и во входном файле. Последняя строка должна заканчиваться символами перевода строки.

**Примеры**

стандартный ввод	стандартный вывод
<pre>10 9 4 ..... .**..... .**..... .....*. .....*. .....*. ..... .....*..... .....*..... .....*.....</pre>	<pre>..... .**..... .**..... ..... .....*** ..... ..... ..... ..... ..... .....</pre>
<pre>10 10 10 ..... ..... ..... ..... .....*..... .....***..... ..... ..... ..... .....</pre>	<pre>..... ..... .....***..... ..... .....*.....*..... .....*.....*..... .....*.....*..... ..... .....***..... .....</pre>

## Задача Е. Литорея

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

Литорея — система тайнописи, применявшаяся в древнерусских текстах. Известно несколько разновидностей литореи, и мы рассмотрим одну из них применительно к текстам из латинских букв.

Текст шифруется с помощью ключа, представляющего собой слово из строчных латинских букв. В шифруемом тексте заменяются только латинские буквы, остальные символы остаются неизменными. Латинские буквы разделяются на блоки так, что длина всех блоков (кроме, может быть, последнего) равна длине ключа. Пусть  $a_1$  — номер первой буквы блока в латинском алфавите,  $b_1$  — номер первой буквы ключа. Тогда первая буква блока заменяется буквой, чей номер в алфавите равен  $a_1 + b_1$  (если  $a_1 + b_1 > 26$ , то берется величина  $a_1 + b_1 - 26$ ). При этом прописная буква заменяется на прописную, а строчная — на строчную. Вторая и последующие буквы блоков шифруются с помощью соответствующих букв ключа аналогичным образом.

Пусть шифруемое слово — «Crusader», а ключом является слово «bow». Тогда первая буква заменяется буквой 'E' (номер первой буквы в латинском алфавите — 3, а номер первой буквы ключа — 2; следовательно, она заменяется буквой с номером 5). Аналогично, вторая буква заменяется буквой 'g' ( $18+15=33$ ,  $33-26=7$ ). Зашифрованный текст, таким образом, выглядит как «Egrupagg». Обратите внимание на то, что в этом примере одинаковые буквы заменяются одинаковыми, однако это случайное совпадение — если расстояние между одинаковыми буквами не кратно длине ключа, такого не произойдет!

Выполните шифрование заданного текста методом литореи.

### Формат входных данных

Входные данные содержат хотя бы одну строку. Первая, непустая строка содержит ключ шифрования, а остальные строки — шифруемый текст (символы с кодами от 32 до 255). Длина одной строки не превышает 30000 символов, а общий объем данных не превосходит 5 мегабайт.

При переходе на следующую строку шифрование продолжается с позиции ключа, следующей за той, с помощью которой было выполнено шифрование предыдущей части текста (как показано во втором примере).

### Формат выходных данных

Количество строк и размер каждой строки выходных данных должны соответствовать шифруемым данным входных, за одним исключением: последняя строка выходных данных должна заканчиваться символами перевода строки, даже если этого нет во входных. Если входные данные содержат только строку с ключом, выведите пустую строку.

### Примеры

стандартный ввод	стандартный вывод
bow Crusader	Egrupagg
rock Deep Purple "Who Do We Think We Are"	Vtha Hjuadt "Zsg Sr Hw Iktfz Zp Sgh"

## Задача F. Мячик на лестнице

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

Ступеньки лестницы пронумерованы сверху вниз, начиная с единицы. Мальчик скатывает мячик с верхней ступеньки так, что он слетает вниз, ударяясь о некоторые из ступенек. Известно, что мячик во время полёта может перелететь через одну или две расположенные ступеньки (а может и удариться о каждую).

Определите минимальное количество промежуточных ударов мячика до того, как он ударится о ступеньку с номером  $N$ .

### Формат входных данных

В единственной строке входных данных записано число  $N$  — целая положительная величина ( $2 \leq N \leq 10^9$ ).

### Формат выходных данных

В единственной строке выведите количество промежуточных ударов.

### Примеры

стандартный ввод	стандартный вывод
5	1
17	5

### Замечание

В первом примере мячик может удариться о любую из ступенек с номерами 2, 3 или 4, перелетая через остальные.