

## Задача 1. Странный банк

Имя входного файла: input.txt или стандартный поток ввода  
Имя выходного файла: output.txt или стандартный поток вывода  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В одном странном банке действуют очень необычные условия по вкладам. Согласно рекламе, проценты по вкладу начисляются ежегодно исходя из "коэффициента вычисления прибыли" (КВП) и зачисляются на отдельный счет (то есть НЕ прибавляются к сумме вклада). КВП изначально равен  $A$  и далее каждый год увеличивается на 1.

В договоре указана точная формула вычисления процентов за год:

$$P = S / (A * (A + 1)),$$

где  $P$  - проценты за год,  $S$  - сумма вклада,  $A$  - КВП на текущий год. Условия по вкладу остаются постоянными в течение всего срока хранения вклада.

Еще одна странность банка заключается в том, что на счете, куда зачисляются проценты, может храниться абсолютно любая сумма, даже если в ней не целое число рублей и копеек (например, на счете может храниться 1/3 рубля).

Кирилл хочет сделать вклад в этом странном банке таким образом, чтобы сумма накопленных процентов оказалась ненулевой. Напишите программу – «калькулятор», которая позволит вычислять сумму процентов, накопившихся за  $N$  лет хранения вклада.

### Формат входного файла

Единственная строка входного файла содержит три целых числа:  $S$  (сумма вклада),  $N$  (количество лет),  $A$  (начальное значение КВП, которое действует в первый год), разделенных пробелами ( $1 \leq S, A \leq 10^9, 1 \leq N \leq 10^{18}$ ).

### Формат выходного файла

Выходной файл должен содержать одно вещественное число, округленное до 3 знаков после запятой – сумму накопленных процентов за  $N$  лет хранения вклада.

### Примеры входных и выходных файлов

input.txt	output.txt
20 2 1	13.333

### Пояснение к примеру

Первый год КВП равнялся 1, поэтому проценты равны  $P = 20 / (1 * 2) = 10$ . На второй год КВП стал равен 2, поэтому проценты за второй год равны  $P = 20 / (2 * 3) = 10 / 3 \approx 3,333$ . В сумме они как раз дают 13.333.

### Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

#### Подзадача 1 (15 баллов)

$$1 \leq S, N, A \leq 1000$$

#### Подзадача 2 (35 баллов)

$$1 \leq S, A \leq 10^9, 1 \leq N \leq 10^6$$

**Подзадача 3 (50 баллов)**

$$1 \leq S, A \leq 10^9, 1 \leq N \leq 10^{18}$$

**Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 2. Международные звонки

Имя входного файла: `input.txt` или стандартный поток ввода  
Имя выходного файла: `output.txt` или стандартный поток вывода  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Топ-менеджер Кирилл отправляется в велосипедное путешествие по Европе!

Конечно, полностью бросать работу он не может, ведь требуется руководить своими подчинёнными. Проблема в том, что международные звонки очень дорогие.

На самом деле, Кирилл не только менеджер, но ещё и немного программист, так что он смог придумать, как ему передавать распоряжения бесплатно.

Как обычно происходит звонок, если абонент А звонит абоненту В?

После начала абонентом А вызова начинается соединение (будем считать, что это занимает  $W$  секунд). Когда соединение установлено, у абонента В начинает звонить телефон, а абонент А слышит гудки. Затем абонент В отвечает на звонок, и начинается разговор.

Дело в том, что плата за исходящий звонок часто рассчитывается с момента начала разговора, а значит, если никто не ответит, то звонок будет бесплатным!

На основании этого факта Кирилл придумал следующий алгоритм:

- Кирилл записывает сообщение, используя только строчные буквы латинского алфавита и символ "пробел".
- Каждый символ имеет численное представление - код в таблице ASCII. Для каждой буквы сообщения Кирилл делает 8 звонков, по одному на каждый бит.
- Если  $i$ -ый бит равен 0, то Кирилл сбрасывает звонок через  $A$  секунд, а иначе через  $B$ .
- Чтобы передать пробел между словами, Кирилл просто не звонит в течение  $S$  секунд.

Например, код символа 'a' - 97. Его битовое представление 0110 0001. Тогда всего на передачу буквы 'a' потратится  $8 * W + (A + B + B + A + A + A + A + B)$  секунд.

Помогите Кириллу рассчитать общее время, которое требуется для передачи его сообщения.

### Формат входного файла

В первой строке даны 4 натуральных числа:  $W, A, B, S$  ( $W, A, B, S \leq 10$ ).

Во второй строке дано единственное натуральное число  $L$  ( $1 \leq L \leq 10^5$ ) – длина сообщения.

В третьей строке дана строка, состоящая ровно из  $L$  символов – собственно сообщение, которое хочет передать Кирилл. Гарантируется, что строка представляет собой последовательность слов, состоящих только из строчных латинских букв, разделённых одним пробелом.

### Формат выходного файла

В единственной строке требуется вывести единственное число – общее время, которое требуется для передачи сообщения.

### Пример входных и выходных файлов

<code>input.txt</code>	<code>output.txt</code>
1 2 3 5 8 a aa aaa	172

***Описание подзадач и системы оценивания***

В этой задаче подзадач нет. Баллы за задачу начисляются только в случае, если все тесты этой задачи успешно пройдены.

***Получение информации о результатах окончательной проверки***

По запросу сообщается результат окончательной проверки на каждом тесте.

### Задача 3. Кирилл и Морской Бой

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Кирилл очень любит менеджмент, но, к сожалению, сейчас у него скучная лекция по математике. Чтобы как-то скрасить своё время наш герой решил поиграть в морской бой. К его сожалению, все кроме него внимательно слушают лекцию, поэтому Кирилл хочет написать программу, которая будет с ним играть.

Кирилл уже реализовал метод, который произвольно расставляет корабли в начале игры и стратегию игры компьютера, но у него возникли проблемы с тем, чтобы отвечать за компьютер: "мимо", "ранил" или "убил" на каждый из выстрелов Кирилла.

Помогите Кириллу и реализуйте для него необходимый метод.

#### **Формат входного файла**

Каждая из первых 10 строк входного файла содержат по 10 символов и задают изначальную расстановку кораблей компьютера, по которым будет стрелять Кирилл.

Символы могут быть только двух видов:

- «.» – Задаёт пустую клетку (клетку в которой нет ни одного корабля)
- «#» – Задаёт клетку, принадлежащую кораблю.

В начальный момент времени все клетки кораблей – целые.

Гарантируется, что никакие два корабля не соприкасаются ни в одной точке.

Гарантируется, что поле содержит 1 корабль длины 4, 2 корабля длины 3, 3 корабля длины 2 и 4 корабля длины 1. Все корабли имеют ширину 1.

Следующая строка содержит единственно целое число  $m$  ( $1 \leq m \leq 100$ ) – количество выстрелов Кирилла.

Далее каждая из  $m$  строк содержит описание очередного выстрела.

Каждый выстрел описывается двумя символами:

- 1) Строчная латинская буква от «a» до «j», задающая столбец атакованной клетки.
- 2) Цифра от 0 до 9, задающая строку атакованной клетки.

Например, команда Кирилла «j0» будет означать выстрел в правую верхнюю клетку.

Гарантируется, что никакие два выстрела не будут совпадать.

#### **Формат выходного файла**

На каждый из  $m$  запросов Кирилла выведите одну из трёх строк:

- «missed» – Если Кирилл очередным выстрелом попал в пустую клетку.
- «damaged» – Если Кирилл попал в корабль, в котором ещё остались целые клетки.
- «killed» – Если Кирилл попал в последнюю целую клетку корабля.

### Пример входных и выходных файлов

input.txt	output.txt
#.....# ...#.#.#... ..... .####...#. .....#. .....#.... .....#.... ##...#...#. ..... ..###..... 6 i3 a9 a0 i2 i4 b7	damaged missed killed missed killed damaged

### Примечание:

Ниже представлено состояние карты кораблей компьютера после выстрелов Кирилла из первого примера:

	a	b	c	d	e	f	g	h	i	j
0	X									o
1				o		o	o			
2									o	
3		o	o	o	o				X	
4									X	
5						o				
6						o				
7	o	X				o			o	
8										
9	o		o	o						

### Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

#### Подзадача 1 (50 баллов)

$$m \leq 10$$

#### Подзадача 2 (50 баллов)

$$m \leq 100$$

### Получение информации о результатах окончательной проверки

По запросу сообщается результат окончательной проверки на каждом тесте.

## Задача 4. Офис на островах

Имя входного файла: `input.txt` или стандартный поток ввода  
Имя выходного файла: `output.txt` или стандартный поток вывода  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Островное государство расположено на  $N$  островах, пронумерованных от 1 до  $N$ . Некоторые острова соединены мостом, общее количество мостов равно  $M$ . Известно, что с любого острова можно проехать на любой другой (возможно, через другие острова), причем единственным способом. К сожалению, проезд по всем мостам является платным: за проезд по  $i$ -му мосту между островами  $v_i$  и  $u_i$  требуется заплатить  $c_i$  рублей. Зато в государстве работает социальное такси, перемещение на нем в пределах одного острова совершенно бесплатно. Также это такси можно заказать для переезда с любого острова на любой другой, стоимость такой поездки будет равна сумме плат за проезд по мостам. Еще для поездок на этом социальном такси можно приобрести годовой абонемент. Такой абонемент позволяет 1 раз в день совершить 1 поездку "туда и обратно", если стоимость поездки в одном направлении не превышает  $X$  рублей. Стоит такой абонемент  $(100 \cdot X)$  рублей, значение  $X$  покупатель выбирает самостоятельно.

Начинающий менеджер Кирилл хочет создать национальную корпорацию по обслуживанию компьютеров, и выбирает подходящее место для размещения офиса. Согласно разработанному бизнес-плану, каждый сотрудник будет выезжать ровно на 1 вызов каждый день, и пользоваться для перемещения социальным такси. Поскольку места вызовов заранее неизвестны, Билл каждому сотруднику приобретет такой годовой абонемент, которого хватит для поездки из офиса на любой остров государства.

Требуется написать программу, которая поможет Биллу выбрать остров для размещения офиса так, чтобы стоимость годового абонемента для сотрудника была минимальной.

### Формат входного файла

Первая строка входного файла содержит два целых числа  $N$  и  $M$ , разделенных пробелом ( $2 \leq N \leq 1000$ ,  $1 \leq M \leq 1000$ ). Далее располагается  $M$  строк, в каждой по три целых числа:  $v_i$ ,  $u_i$ ,  $c_i$ , разделенных пробелами ( $1 \leq v_i, u_i \leq N$ ,  $1 \leq c_i \leq 10^5$ ). Одна строка описывает один мост: какие острова он соединяет и стоимость проезда по нему.

### Формат выходного файла

Выходной файл должен содержать одно целое число – номер острова для размещения офиса. Если для нескольких островов стоимость абонементов одинаковая, вывести наименьший из номеров.

### Примеры входных и выходных файлов

<code>input.txt</code>	<code>output.txt</code>
4 3 1 2 1 2 3 2 4 1 5	1
5 4 1 4 10 5 4 10 2 4 5 2 3 10	2

### **Описание подзадач и системы оценивания**

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

#### **Подзадача 1 (30 баллов)**

$N \leq 200$ , острова расположены вдоль одной прямой, мосты соединяют все пары соседних островов

#### **Подзадача 2 (30 баллов)**

$N \leq 200$

#### **Подзадача 3 (40 баллов)**

$N \leq 2000$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.



## Задача 5. Кирилл и Лампочки

Имя входного файла: `input.txt` или стандартный поток ввода  
Имя выходного файла: `output.txt` или стандартный поток вывода  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Кирилл уже отдал все указания своим подчинённым менеджерам и теперь может пойти домой развлекаться с ООП. Но перед уходом он должен погасить все лампочки на его приборной панели. Лампочки расположены в квадрате из  $n$  рядов по  $n$  лампочек.

Напротив каждого ряда и каждого столбца есть переключатели, которые меняют состояния всех целых лампочек в этом ряду или столбце соответственно. На использование одного переключателя Кирилл потратит одну секунду. Но наш топ-менеджер очень хочет скорее окунуться в мир ООП, поэтому ради такого он даже готов ломать лампочки. За одну секунду Кирилл может сломать одну любую лампочку, и после этого она перестанет гореть навсегда.

Помогите Кириллу уйти домой пораньше и выдайте ему инструкции, чтобы он смог погасить все лампочки за минимальное время.

### Формат входного файла

Первая строка входного файла содержит число  $n$  ( $2 \leq n \leq 20$ ). Следующие  $n$  строк задают начальные состояния лампочек. Строка с номером  $i$  содержит  $n$  символов. Если символ в  $i$ -й строке на позиции  $j$  равен единице, то  $j$ -я лампочка в строке с номером  $i-1$  изначально горит, иначе нет.

### Формат выходного файла

Первая строка выходного файла должна содержать единственное целое неотрицательное число  $t$  – наименьшее количество секунд, за которое Кирилл сможет погасить все лампочки.

Следующие  $t$  строк должны описывать  $t$  инструкций для Кирилла. Они должны быть одного из трёх видов:

- $R\ y$  – переключить все целые лампочки в строке с номером  $y$ .
- $C\ x$  – переключить все целые лампочки в столбце с номером  $x$ .
- $V\ y\ x$  – в строке с номером  $y$  сломать лампочку на позиции  $x$ .

При наличии нескольких оптимальных ответов разрешается вывести любой из них.

### Пример входных и выходных файлов

<code>input.txt</code>	<code>output.txt</code>
4 0010 0010 0010 1111	3 R 4 V 4 3 C 3
2 11 11	2 R 2 R 1

### Примечания

В каждом из приведённых примеров могут быть и другие последовательности оптимальных инструкций. Например, во втором примере Кирилл мог бы погасить все лампочки за 2 хода следующим набором команд: “C 1” и “C 2”.

ООП – Объектно-Ориентированное Программирование. Одно из любимых занятий Кирилла.

### **Описание подзадач и системы оценивания**

Баллы за каждую подзадачу начисляются только в случае, если все тесты этой подзадачи успешно пройдены.

#### **Подзадача 1 (20 баллов)**

$$n \leq 4$$

#### **Подзадача 2 (20 баллов)**

$$n \leq 8$$

#### **Подзадача 3 (20 баллов)**

$$n \leq 12$$

#### **Подзадача 4 (20 баллов)**

$$n \leq 16$$

#### **Подзадача 5 (20 баллов)**

$$n \leq 20$$

### **Получение информации о результатах окончательной проверки**

По запросу сообщается результат окончательной проверки на каждом тесте.