

## Задача 1. Алфавит

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Совсем недавно Лёша начал изучать в школе английский язык. Как это часто бывает, в некоторых аспектах изучения этого предмета он достиг непревзойдённых высот, а в других, наоборот, столкнулся с определенными проблемами. Так, например, Лёша уже успел выучить первые  $P$  букв латинского алфавита. Теперь перед ним стоит непростая задача — нужно научиться красиво писать эти буквы, с чем у него как раз и возникли трудности.

Преподаватель рассказал своему ученику очень эффективную схему для улучшения качества написания букв. Сначала нужно выписать в тетрадь все известные буквы в алфавитном порядке. Далее нужно сделать то же самое, но только выписывать каждую букву по два раза подряд. На следующем этапе нужно поступать точно так же, только выписывать буквы по три подряд. Так продолжать, пока не станет хорошо получаться.

Таким образом, если, например, Лёша выучил первые три буквы алфавита, то в тетрадь он будет выписывать строку следующего вида: `abcaabbccaaabbbcccaaaaabbbbcccc ...`

Естественно, одни буквы Лёше нравятся больше, другие меньше. Как оказалось, ему нравятся только те буквы, которые встречаются на обложке учебника по английскому языку.

Лёше требуется написать в тетрадь строку заданного вида длиной в  $R$  букв. Он уже выписал  $(L - 1)$  букву. И теперь его интересует вопрос: сколько ещё букв, которые ему нравятся, предстоит ему выписать. Напишите программу, позволяющую ответить на вопрос, который тревожит Лёшу.

### Формат входного файла

В первой строке входного файла записано одно целое число:  $P$  — количество букв, которые Лёша уже успел выучить ( $1 \leq P \leq 26$ ).

Во второй строке записано целое число  $K$  — количество букв, изображённых на обложке учебника ( $1 \leq K \leq 50$ ).

В следующей строке записаны через пробел ровно  $K$  маленьких букв латинского алфавита, которые изображены на обложке учебника.

В четвёртой строке записаны два целых числа —  $L$  и  $R$  ( $1 \leq L \leq R \leq 10^{18}$ ).

### Формат выходного файла

В выходной файл требуется вывести одно целое число — количество букв, которые Лёше нравятся, и которые ещё осталось ему выписать.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
4 3 a b d 1 4	3
2 4 a c c a 8 11	2

## Комментарий

Во втором тестовом примере Лёша выучил всего две буквы –  $a$  и  $b$ . Среди букв, которые напечатаны на обложке учебника, встречается из выученных только буква  $a$ . После того, как Лёша допишет все 11 букв, в тетради окажется следующая строка: *abaabbaabb*. Среди дописанных букв две буквы  $a$ , которые и нужно посчитать.

## Задача 2. Наперстки

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Азартные игры дело рискованное, но у Гриши есть план!

Гриша играет с дедушкой в наперстки. На столе стоят перевернутые  $N$  колпачков, под одним из них спрятан шарик. Каждую минуту Гриша может поднять и проверить один колпачок. При этом в некоторые моменты времени дедушка может поменять некоторые наперстки местами. Это происходит так быстро, что Гриша забывает, какие из переставленных колпачков он проверил.

Если же дедушка меняет местами только те наперстки, под которыми по Гришиным сведениям к этому моменту точно нет шарика, то такого факта он не забывает. Все перемешивания происходят не более одного раза в минуту.

В одну и ту же минуту могут быть сделаны и перестановка колпачков дедушкой, и проверка Гриши. В этом случае сначала происходит перестановка, а потом проверка. Считается, что игра начинается с минуты под номером 0.

Помогите Грише посчитать, сколько минут ему потребуется, чтобы найти шарик.

### Формат входного файла

В первой строке входного файла записаны через пробел два целых числа  $N$  и  $M$ , где  $N$  – количество колпачков, а  $M$  – количество перестановок ( $1 \leq N \leq 1000, 0 \leq M \leq 10$ ). Колпачки пронумерованы числами от 1 до  $N$ .

В следующих  $M$  строках описываются перестановки, каждая на отдельной строке. Описание каждой перестановки начинается с целого числа  $T$  – номера минуты, на которой произошла перестановка колпачков. Далее через пробел записано число  $L$  – количество переставленных колпачков, затем записано  $L$  целых чисел – номера переставленных колпачков ( $0 \leq T \leq 10000, 1 \leq L \leq N$ ). Описание перестановок дается в произвольном порядке.

### Формат выходного файла

В выходной файл необходимо вывести одно целое неотрицательное число – минимальное количество минут, которые требуются, чтобы гарантированно найти шарик.

### Пример

<code>input.txt</code>	<code>output.txt</code>
10 2 3 2 1 8 5 7 1 2 3 4 5 6 7	12

### Комментарий

В примере последовательность осмотров колпачков может быть, например, такой:  
1 8 10 9 – 1 2 3 4 5 6 7.

Знак «минус» означает, что на пятой минуте проверка пропущена.

## Задача 3. Электрички

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

По выходным дням многие горожане стремятся выехать из пыльного города на дачу. Учитывая, что по утрам в субботу очень много людей скапливается на главном вокзале, диспетчерская служба пригородного железнодорожного транспорта разработала специальное расписание. Все электрички в это время идут только из города в сторону дачных поселков. В обратную сторону движения нет.

Система дорог в этом регионе устроено очень своеобразно. На любую станцию можно приехать только из одного определенного населенного пункта, выехать можно не более чем по двум направлениям. При выезде со станции расположена стрелка. Она открывает проезд в одно направление и закрывает на второе. Если нужно поезду ехать в сторону перекрытого направления, то приходится эту стрелку переключать машинисту вручную. Город не является исключением. Из его главного вокзала можно выехать не более чем на две другие станции. Нет ни одной станции, на которую можно вернуться, выехав из нее.

Согласно разработанному расписанию, поезда уходят из города один за другим. Для каждого из них указан пункт назначения. Диспетчеры хотят выяснить, какое минимальное количество раз за день придется машинистам переводить стрелки при заданном расписании поездов.

### Формат входного файла

В первой строке входного файла записаны через пробел три числа  $N$ ,  $M$  и  $R$  – общее количество станций, количество поездов и номер главного вокзала ( $1 \leq N, R \leq 10^3, 1 \leq M \leq 10^4$ ). Все станции, включая городской главный вокзал, пронумерованы числами от 1 до  $N$ .

Следующие  $N - 1$  строк содержат описание перегонов между станциями в формате  $UVT$ , где  $U$  и  $V$  – номера станций, которые соединяет этот перегон от станции  $U$  в направлении станции  $V$ , и  $T$  – состояние стрелки при выезде со станции  $U$ . Если  $T = 1$ , то это означает, что стрелка повернута в сторону станции  $V$ , если  $T = 0$ , то путь в сторону станции  $V$  закрыт.

Последняя строка содержит расписание движения пригородных поездов. Это  $M$  чисел, записанных через пробел, которые задают номера конечных станций, в порядке выезда электричек с главного вокзала.

### Формат выходного файла

В выходной файл необходимо вывести одно целое неотрицательное число – минимальное количество переводов стрелок.

### Пример

<code>input.txt</code>	<code>output.txt</code>
7 4 6 6 7 1 6 3 0 3 4 1 3 2 0 2 5 1 2 1 0 7 4 5 1	3

## Задача 4. Окружность

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

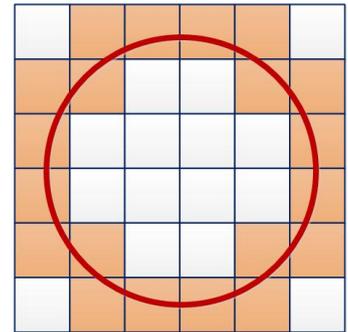
На уроке геометрии учились рисовать циркулем. У Пети тетрадка в клеточку. Он поставил основание циркуля в угол одной из клеточек, отмерил по прямой, проходящей через этот угол, несколько клеточек и нарисовал окружность с радиусом, равным количеству отмеренных клеточек.

Его заинтересовал вопрос – сколько клеточек пересекает эта окружность? На листочке бумаги он такие клетки закрасил и легко посчитал их количество.

Но как определить это число, не рисуя окружность, и зная не радиус, а длину окружности?

Помогите Пете решить эту задачу. Напишите программу, которая по заданной длине окружности с центром в точке пересечения двух линий, образующих клетки, определяет, сколько клеток она пересекает.

Будем предполагать, что граница клетки является ее частью. Считается, что окружность проходит через клетку, если клетка содержит хотя бы одну точку окружности. Каждая клетка – это квадрат со стороной, равной 1.



### Формат входного файла

Во входном файле записано одно целое число  $L$  – длина окружности ( $0 < L < 10^{18}$ )

### Формат выходного файла

В выходной файл нужно вывести одно целое число – количество клеточек, которые заданная окружность пересекает.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
1	4
7	12

## Задача 5. Выставка картин

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Художник к своей юбилейной выставке нарисовал  $N$  картин. Первую нарисованную картину он оценил в  $A$  рублей. Стоимость каждой вновь нарисованной картины вычислялась как сумма некоторого числа  $D$  и стоимости предыдущей картины.

Выставка прошла с огромным успехом. Галерея выразила желание приобрести любые  $K$  произведений художника из тех  $N$  картин, которые он представил публике.

Какую максимальную сумму денег может выручить художник от продажи  $K$  картин?

### Формат входного файла

В первой строке входного файла записано через пробел два целых числа  $N$  и  $K$ , где  $N$  – количество всех картин, а  $K$  – количество картин, которые хочет купить галерея ( $0 \leq K \leq N \leq 1000$ ).

Во второй строке заданы еще два числа – стоимость первой нарисованной картины и  $D$  – разность между стоимостями двух подряд нарисованных картин ( $-1000 \leq A \leq 1000, -1000 \leq D \leq 1000$ ).

### Формат выходного файла

В выходной файл необходимо вывести одно целое число – максимальную сумму денег, которую можно выручить от продажи  $K$  картин.

### Пример

<code>input.txt</code>	<code>output.txt</code>
5 3 1 1	12