

## Задача А. Подарок Диппера

Имя входного файла: `changestr.in`  
Имя выходного файла: `changestr.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Во время очередного приключения Диппер нашел строку  $s$  длины  $n$ . Он считает, что эта строка является идеальным подарком для Мэйбл. Она привередливая, поэтому не каждая строка ей понравится. К счастью, у Диппера есть знакомый мастер, который умеет изменять строки за определенное количество монет.

Мальчик хочет угодить Мейбл и сделать строку, которая ей понравится, потратив минимальное количество монет. Мастер имеет каталог из  $m$  операций замены. Каждая операция позволяет заменить определенный символ  $a$  в любой позиции строки на символ  $b$ , заплатив  $c$  монет. Любую операцию можно использовать неограниченное количество раз в любой позиции строки. Мастер может заменять символы, которые он сам раньше ставил на эту позицию. В каталоге мастера может быть несколько операций изменение  $a$  на  $b$  с разными стоимостями.

Строка называется  $k$ -строкой, если она может быть представлена в виде  $k$  копий некоторой строки, записанных подряд. Например, строка «`aabaabaabaab`» является одновременно 1-строкой, 2-строкой и 4-строкой, но не является 3-строкой, 5-строкой, 6-строкой и так далее. Назовем строку «красивой», если она является  $k$ -строкой, для  $k$  больше единицы. Мейбл нравятся только красивые строки. Помогите Дипперу понять, может ли он получить красивую строку, а если может, то какое минимальное количество монет ему необходимо потратить на работу мастера.

### Формат входных данных

В первой строке заданы числа  $n$  и  $m$  — длина строки  $s$  и количество операций ( $2 \leq n \leq 10^5$ ;  $1 \leq m \leq 10^5$ ).

Во второй строке задана последовательность маленьких латинских букв длины  $n$  — строка  $s$ .

Далее следует  $m$  строк. В каждой записаны две маленькие латинские буквы  $a$ ,  $b$  и число  $c$  — операция, которая соответствует замене символа  $a$  на  $b$  за цену  $c$  ( $0 \leq c \leq 100\,000$ ).

### Формат выходных данных

Если не существует способа сделать строку  $s$  красивой, то выведите  $-1$ , иначе выведите количество монет, которое нужно потратить.

### Пример

<code>changestr.in</code>	<code>changestr.out</code>
<code>6 4</code> <code>abcdba</code> <code>d a 3</code> <code>a z 3</code> <code>z c 2</code> <code>a d 1</code>	<code>6</code>

### Замечание

`abcdba` → `dbcdba` → `dbcdbz` → `dbcdbc`

1) Заменяем букву `a` на `d`, заплатив 1.

2) Заменяем букву `a` на `z`, заплатив 3.

3) Заменяем букву `z` на `c`, заплатив 2.

Ответ:  $1 + 3 + 2 = 6$

Строка `dbcdbc` является 2-строкой.

## Задача В. Шкаф для обуви

Имя входного файла: `cupboard.in`  
Имя выходного файла: `cupboard.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В семье Мэйбл появился новый шкаф для обуви, который довольно быстро заполнился, так как семья у них немаленькая, да и гостей довольно много. Мэйбл и тут проявила активность и пристально следила за тем, как люди ставят и забирают обувь из шкафа.

Тем временем шкаф оказался довольно необычный — полки в нем расположены на разных высотах, а именно первая полка расположена на высоте  $h_1$  от пола, вторая — на высоте  $h_2$ , ...,  $n$ -я полка расположена на высоте  $h_n$ . Достаточно изучив, как люди кладут и забирают обувь, Мэйбл заметила, что человек с ростом  $height$  не может поставить свою пару обуви на полку высотой больше  $height$ , а также на полку высотой меньше  $height/k$ . Также она заметила, что у человека с ростом  $height$  размер ноги явно не меньше  $height/m_1$  и не больше  $height/m_2$ . Теперь Мэйбл смотрит на шкаф и ей кажется, что некоторые пары обуви появились там волшебным образом — ни один человек не мог их туда поставить, исходя из ее наблюдений. Помогите ей проверить это — посчитайте количество пар обуви, которые не удовлетворяют наблюдениям.

### Формат входных данных

В первой строке дано число  $n$  — количество полок в шкафу ( $1 \leq n \leq 10^5$ ).

Во второй строке дано три числа  $k$ ,  $m_1$  и  $m_2$  — коэффициенты, выведенные Мэйбл ( $1 \leq k \leq 100$ ;  $1 \leq m_2 \leq m_1 \leq 100$ ).

В следующих  $n$  строках дано описание полок. В начале описания  $i$ -й полки надо число  $h_i$  — ее высота ( $1 \leq h_1 \leq 10^7$ ,  $h_{i-1} < h_i \leq 10^7$  для всех  $i > 1$ ). Далее дано число  $k_i$  — количество пар обуви на этой полке ( $1 \leq k_i \leq 10^5$ ). После этого дано  $k_i$  чисел  $size_{i,j}$  — размеры пар обуви на этой полке ( $1 \leq size_{i,j} \leq 10^7$ ).

Гарантируется, что сумма всех  $k_i$  не превосходит  $10^5$ .

### Формат выходных данных

В единственной строке выведите количество пар обуви, которые не удовлетворяют наблюдениям Мэйбл.

### Пример

<code>cupboard.in</code>	<code>cupboard.out</code>
3	2
2 1 1	
1 2 1 2	
2 2 1 4	
4 2 3 4	

### Замечание

В тестовом примере всего две пары обуви, не удовлетворяющие наблюдениям Мэйбл — пара размером 1 на полке высотой 2 и пара размером 3 на полке высотой 4.

## Задача С. Цифровая загадка

Имя входного файла: `digits.in`  
Имя выходного файла: `digits.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Распутывая очередную тайну, Диппер наткнулся на непростую загадку.

На доске написано  $n$  чисел. Также можно взять любую цифру любого числа и заменить на любую другую. Однако, эту операцию можно выполнить не более  $k$  раз.

Дипперу нужно, чтобы сумма записанных чисел была максимально возможной. Он хочет узнать наибольшее число, на которое он сможет увеличить сумму применением данной операции. Помогите ему найти разгадку!

### Формат входных данных

В первой строке входного файла даны два целых числа  $n, k$  — количество чисел на доске и ограничение на количество операций. ( $1 \leq n \leq 1000, 1 \leq k \leq 10^4$ )

Во второй строке записано  $n$  чисел  $a_i$  — числа на доске. ( $1 \leq a_i \leq 10^9$ )

### Формат выходных данных

В выходной файл выведите единственное число — разность между суммами после применения операции и начальной.

### Примеры

<code>digits.in</code>	<code>digits.out</code>
5 2 1 2 1 3 5	16
3 1 99 5 85	10
1 10 9999	0

### Замечание

В первом примере можно заменить единицы на девятки, тогда сумма изменится на 16.

Во втором примере можно заменить 85 на 95, и сумма увеличится на 10.

В третьем примере у числа 9999 нельзя заменить хотя бы одну цифру так, чтобы число увеличилось, поэтому ответ 0.

## Задача D. Зеркало

Имя входного файла: `mirror.in`  
Имя выходного файла: `mirror.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Стэн хочет поставить зеркало в своём офисе, и сейчас он выбирает место для него.

Офис Стэна — это комната, имеющая форму выпуклого многоугольника. Стэн собирается поставить зеркало так, чтобы оно занимало какую-то стену целиком, но он ещё не определился, какую именно.

Стэн считает, что площадь той части комнаты, отражение которой он будет видеть в зеркале со своего рабочего места, является важной характеристикой этого зеркала. Помогите Стэну посчитать эту площадь для каждого возможного положения зеркала.

### Формат входных данных

В первой строке входного файла находится число  $n$  — количество вершин многоугольника ( $3 \leq n \leq 10^5$ ).

В следующих  $n$  строках заданы координаты вершин многоугольника в порядке обхода против часовой стрелки.

В последней строке заданы координаты рабочего места Стэна.

Все координаты целые и не превосходят  $10^8$  по абсолютному значению. Гарантируется, что многоугольник выпуклый и никакие три вершины не лежат на одной прямой, а рабочее место Стэна находится строго внутри него.

### Формат выходных данных

Для каждой стороны многоугольника выведите на отдельной строке площадь части комнаты, отражение которой будет видно с рабочего места Стэна, если зеркало поставить на месте этой стороны.

Рёбра упорядочены в порядке обхода против часовой стрелки, начиная с первой вершины.

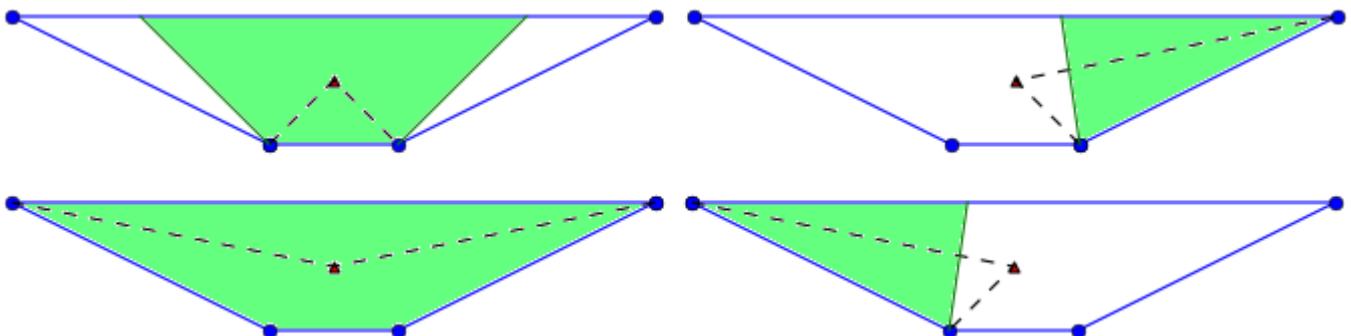
Ответ будет считаться правильным, если его абсолютная или относительная погрешность не превосходит  $10^{-6}$ .

### Пример

<code>mirror.in</code>	<code>mirror.out</code>
4	8.0000000
-1 -1	4.2857143
1 -1	12.0000000
5 1	4.2857143
-5 1	
0 0	

### Замечание

Пояснение к примеру: зелёным цветом выделена область, отражение которой видно в зеркале, для каждой из четырёх сторон; пунктиром — отрезки от рабочего места Стэна до границ зеркала.



## Задача E. В поисках неизведанного

Имя входного файла: `paths.in`  
Имя выходного файла: `paths.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Диппер и Мэйбл решили еще раз обследовать Гравити Фолз. Надо сказать, что Гравити Фолз не сильно отличается по общему устройству от других городов — он также представляет из себя совокупность домов, соединенных улицами. Причем для каждой пары домов существует не более одной улицы их соединяющей. Да и петель, то есть улиц, соединяющих дом с самим собой, тоже нет. Также известно, что если по улице можно добраться от дома А до дома В, то и от дома В до дома А можно добраться по этой же улице.

Сейчас Диппер и Мэйбл решили составить список маршрутов, которые бы посещали каждый дом ровно один раз. (То есть если в городе  $n$  домов, то в маршруте будет ровно  $n$  различных чисел — номеров домов, и между любыми двумя соседними будет существовать одна улица). Диппер и Мэйбл считают два маршрута разными, если в них разные последовательности домов.

Список оказался довольно большим. К тому же Диппер и Мэйбл не уверены, что он правильный. Для того чтобы проверить выкладки, они хотели бы для начала знать количество таких путей. Без Вас им точно не обойтись!

### Формат входных данных

В первой строке входного файла дано натуральное число  $n$  — количество домов ( $1 \leq n \leq 1000$ ). Далее следуют  $n$  строк. Каждая  $i$ -ая строка задана в следующем формате: первое число в строке  $k$  — число соседних (то есть связанных улицей) домов для  $i$ -го дома, далее перечислены  $k$  различных чисел — номера соседних с  $i$ -ым домов.

### Формат выходных данных

Выведите одно число — количество вышеописанных маршрутов. Поскольку данное число может быть довольно большим, выведите его по модулю 2.

### Пример

<code>paths.in</code>	<code>paths.out</code>
5	0
3 2 3 5	
3 1 3 4	
4 1 2 4 5	
3 2 3 5	
3 1 3 4	

## Задача F. Тайная комната

Имя входного файла: `secretroom.in`  
Имя выходного файла: `secretroom.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В процессе раскрытия очередного секрета городка Гравити Фолз близнецам Дипперу и Мэйбл пришлось отправиться в таинственный лес, в чаще которого они набрали на заброшенный дом. Дом был очень старым, и близнецы не обнаружили в нем ничего примечательного, за исключением потайной комнаты, располагавшейся в подвальной части дома. Комната оказалась заперта, а на двери висел домофон.

Диппер твердо решил во что бы то ни стало открыть комнату. В этот раз ему повезло — на одной из страниц дневника он нашел ключ к разгадке кода, который необходимо ввести на домофоне, чтобы открыть дверь. В дневнике была записана некоторая последовательность чисел  $a$  длины  $n$  и говорилось, что кодом является максимальная по количеству чисел её подпоследовательность, для каждой пары чисел которой выполняется следующее неравенство:  $a_i - a_j < j - i$ . Помогите Дипперу найти длину кода.

### Формат входных данных

В первой строке входного файла дано число  $n$  — количество элементов последовательности  $a$  ( $1 \leq n \leq 10^6$ ).

В следующей строке даны элементы последовательности — целые неотрицательные числа  $a_i$  ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите единственное число — количество чисел в коде, с помощью которого Диппер сможет открыть дверь тайной комнаты.

### Примеры

<code>secretroom.in</code>	<code>secretroom.out</code>
3 1 2 2	3
6 5 3 5 6 6 5	4

### Замечание

В первом тестовом примере  $a_0 = 1, a_1 = 2, a_2 = 2$ . Рассмотрев все пары чисел, можно убедиться, что для каждой из них неравенство выполняется:

- 1)  $a_0 - a_1 = 1 - 2 < 1 - 0$ .
- 2)  $a_1 - a_2 = 2 - 2 < 2 - 1$ .
- 3)  $a_0 - a_2 = 1 - 2 < 2 - 0$ .

Во втором примере неравенства будут выполняться, например, при выборе чисел с индексами 1, 2, 3 и 4. При большем количестве чисел найдется такая пара, для которой неравенство выполняться не будет.

## Задача G. Починка хижины

Имя входного файла: `squares.in`  
Имя выходного файла: `squares.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Во время Странногеддона Хижина чудес была довольно сильно повреждена. Лето закончилось, наступила осень, стало холодать, и жить с дырой в стене Стэну стало совершенно невыносимо.

К счастью, в подвале нашелся прямоугольный кусок странного неземного металла (вероятно, утащенный с летающей тарелки)  $n$  метров в высоту и  $m$  метров в ширину. Суз сказал, что проще всего нарезать из этого куска металла несколько квадратных кусков размером  $k$  на  $k$  метров и заклеить ими все дыры. Так бы и сделали, но определить с размером этого куска оказалось непросто — каждый голосует за свой вариант размера квадрата. Все, на чем удалось договориться, так это то, что длины всех квадратов должны быть равны какому-то одному и тому же целому числу  $k$ , и для этого  $k$  нужно вырезать как можно больше квадратов со стороной  $k$  из исходного куска металла. Для простоты проводить разрезы можно только параллельно сторонам листа металла, а каждый квадрат не должен содержать внутри себя ни одного разреза (или быть составленным из нескольких меньших кусков металла).

Поскольку Диппер и Мэйбл уехали, некому помочь Стэну выбрать размер стороны квадрата. Прежде чем приниматься за работу, Стэн решил для каждого  $k$  посчитать, какое наибольшее количество квадратов со стороной  $k$  можно вырезать из прямоугольника  $n$  на  $m$ . Поскольку вариантов  $k$  слишком много, его удовлетворит сумма этих чисел для всех возможных  $k$ . Помогите ему посчитать это значение.

### Формат входных данных

В первой строке даны два целых числа  $n, m$  — высота и ширина прямоугольного куска металла ( $1 \leq n, m \leq 10^9$ ).

### Формат выходных данных

Выведите одно целое число — сумму по всем значениям  $k$  количества квадратов  $k$  на  $k$ , которое можно вырезать из прямоугольника  $n$  на  $m$ .

### Примеры

<code>squares.in</code>	<code>squares.out</code>
3 4	15
7 7	66

### Замечание

В первом тестовом примере из прямоугольника 3 на 4 можно вырезать 12 квадратов со стороной 1, два квадрата со стороной 2, и один квадрат со стороной 3. Итого,  $12+2+1=15$ .

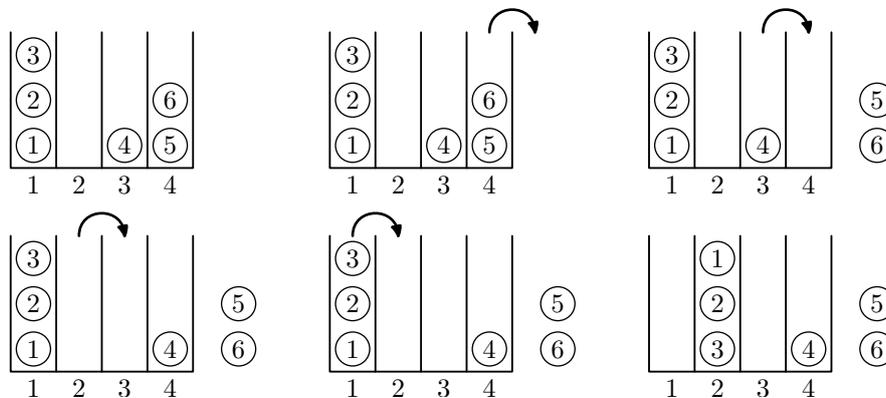
## Задача Н. Очередь к аттракциону

Имя входного файла:	queue.in
Имя выходного файла:	queue.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сегодня прадея Стэн устроил ярмарку с аттракционами. Диппера заинтересовал один аттракцион, но, к сожалению, чтобы на нем прокатиться, нужно сыграть в игру от прадеды и затем отстоять длинную очередь.

Игра (кто вообще придумал ее так назвать?) происходит следующим образом: люди стоят в  $n$  колонн (колонны нумеруются слева направо), затем происходят сдвиги. В процессе сдвига последняя колонна, начиная с первого человека, выходит из игры и уходит в очередь к аттракциону, предпоследняя колонна, начиная с первого человека, переходит в последнюю колонну, при этом первый человек становится последним, и так далее, все остальные колонны таким же образом сдвигаются на одну вперед.

Например, вот так произойдет сдвиг для такого расположения людей:



В результате люди с номерами 5 и 6 уходят в очередь на аттракцион, а остальные продолжают играть.

Когда новый человек приходит, чтобы участвовать в игре, он становится первым (люди в колонне нумеруются сверху вниз) в самую левую колонну. Сейчас Диппер стоит рядом с левой колонной, а очередь к аттракциону еще пуста. Ровно через  $t$  минут произойдет первый сдвиг, а затем они будут происходить спустя каждые  $t$  минут. Дитя Времени сообщило Дипперу времена, в которые подойдут все оставшиеся люди, желающие принять участие в игре и прокатиться на аттракционе. Причем никакие два человека не подойдут в одну минуту. Теперь он размышляет, как лучше поступить, чтобы прокатиться на аттракционе раньше всего. А именно, он решил, что пропустит перед собой некоторое, возможно нулевое, количество еще не подошедших людей, и после этого сразу же войдет в игру, то есть встанет в первую колонну. Если он решит никого не пропускать, он встанет в игру прямо сейчас. При этом он хочет, чтобы количество людей, которые прокатятся на аттракционе перед ним, было как можно меньше.

Считайте, что сдвиг происходит мгновенно. Если в одну и ту же минуту подходит человек и происходит сдвиг, человек успевает войти в игру перед сдвигом. При этом Диппер также может войти в игру после этого человека и перед сдвигом.

Помогите Дипперу, выясните, какое минимальное количество человек успеет прокатиться на аттракционе перед ним, и сколько человек он должен для этого пропустить вперед. Если он может пропустить разное количество людей, выведите минимальное из них.

### Формат входных данных

В первой строке находятся три целых числа:  $n$ ,  $m$  и  $t$  — количество колонн, людей, которые подойдут позже, и время между сдвигами соответственно ( $1 \leq n, t \leq 100\,000$ ;  $0 \leq m \leq 100\,000$ ).

Во второй строке содержится  $n$  целых чисел  $a_i$  — количество людей в колоннах в текущий момент ( $0 \leq a_i \leq 100\,000$ ).

В третьей строке содержится  $m$  целых чисел  $t_i$  — количество минут, спустя которые подойдет  $i$ -й человек ( $1 \leq t_i \leq 100\,000$ ;  $t_i < t_{i+1}$ ).

### Формат выходных данных

В единственной строке выведите два числа: минимальное количество человек, которые прокатятся на аттракционе перед Диппером, и количество человек, которых он должен пропустить вперед.

Если ответ не единственен, выведите тот, в котором второе число минимально.

### Пример

queue.in	queue.out
4 3 4	6 0
3 0 1 2	
1 4 6	

## Задача I. Диппер и аппарат

Имя входного файла: `stringsqueries.in`  
Имя выходного файла: `stringsqueries.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Сегодня Диппер нашел на чердаке странный аппарат. Он содержит  $n$  слотов, пронумерованных от 1 до  $n$  и расположенных подряд. В каждом слоте находится строка. Изначально во всех слотах находятся пустые строки. Аппарат может по заданным  $l$ ,  $r$  и  $s$  добавить в конец всех строк, находящихся с  $l$ -го по  $r$ -й слот, строчку  $s$ .

Диппер поспорил с Мэйбл, что она не сможет смоделировать действия аппарата. Для этого он будет давать команды, а Мэйбл будет параллельно их повторять. Для проверки того, что Мэйбл безошибочно повторяет поведение аппарата, Диппер будет иногда спрашивать у Мэйбл следующий вопрос: чему равна подстрока с  $x$  по  $y$  в строке, находящейся в слоте  $i$ .

Мэйбл решила написать программу, которая будет моделировать этот процесс. Она просит вас помочь ей.

### Формат входных данных

В первой строке находятся два натуральных числа  $n$ ,  $m$  — количество слотов в аппарате и количество команд Диппера ( $1 \leq n \leq 2 \cdot 10^5$ ;  $1 \leq m \leq 2 \cdot 10^5$ ).

В следующих  $m$  строках находятся  $m$  команд по одной в строке, каждая команда может быть одного из двух типов:

- $1\ l\ r\ s$  — добавить в конец всех строк, находящихся с  $l$ -го по  $r$ -й слот включительно, непустую строку  $s$  ( $1 \leq l \leq r \leq n$ ). Строка  $s$  состоит из строчных латинских букв.
- $2\ i\ x\ y$  — узнать значение подстроки с  $x$  по  $y$  включительно у строки, находящейся в слоте  $i$  ( $1 \leq i \leq n$ ;  $1 \leq x \leq y$ ). Гарантируется, что длина строки, находящейся в слоте под номером  $i$ , не меньше, чем  $y$ .

Гарантируется, что сумма длин строк, входящих в команды первого типа, не превосходит  $10^6$ .

Гарантируется, что сумма длин подстрок по всем командам второго типа не превосходит  $10^6$ .

### Формат выходных данных

На каждую команду второго типа нужно вывести ответ в отдельной строке.

### Примеры

<code>stringsqueries.in</code>	<code>stringsqueries.out</code>
4 4 1 2 2 two 1 2 3 aa 2 3 1 1 2 2 1 4	a twoa
3 6 1 1 2 ab 1 2 3 cd 2 2 2 4 2 3 1 2 1 1 3 xyzu 2 1 2 5	bcd cd bxyz