

## Содержание

Задача А. $A + B$ [2 секунды, 256 МиБ]	2
Задача В. Капча [2 секунды, 256 МиБ]	3
Задача С. Электрическая цепь [2 секунды, 256 МиБ]	5
Задача D. Очень важная формула [2 секунды, 256 МиБ]	6
Задача Е. Бубновый джокер [2 секунды, 256 МиБ]	7
Задача F. Лекция [2 секунды, 256 МиБ]	8
Задача G. Солдаты-самоубийцы [2 секунды, 256 МиБ]	9
Задача H. Квadratura круга [2 секунды, 256 МиБ]	10
Задача I. Расписание [2 секунды, 256 МиБ]	11
Задача J. Изгрызенные книги [2 секунды, 256 МиБ]	12

## Задача А. А + В [2 секунды, 256 МиБ]

— Не комплексуй!

из письма п к і

Когда один известный алгебраист (далее — О. И. А.) с мат-меха был маленьким, ему в школе строго-настрого запретили брать квадратные корни из отрицательных чисел. Конечно же, О. И. А. на запрет наплевал и даже (о, ужас!) стал использовать то, что получается после взятия такого корня, в качестве основания системы счисления.

Недавно О. И. А. нашёл свои старые записи, а в них какое-то подозрительное тождество. Чтобы проверить его справедливость, О. И. А. достаточно сложить два числа и посмотреть на результат. Но, к сожалению, эти два числа записаны как раз в системе счисления с основанием, равным корню из отрицательного числа (а конкретно,  $\sqrt{-4}$ ), и ответ тоже должен быть представлен в системе счисления с основанием  $\sqrt{-4}$ . Ваша задача — помочь О. И. А. проверить тождество, сложив два числа.

Как известно, *мнимые числа* — это как раз те числа, которые получаются в результате взятия квадратного корня из отрицательного числа. В *системе с мнимым основанием*  $\sqrt{-k}$  в числах используются цифры от 0 до  $k-1$ , включительно. Мнимые основания систем счисления ведут себя так же, как обычные, то есть число, записанное как  $d_3d_2d_1d_0$ , вычисляется по формуле  $d_3 \cdot b^3 + d_2 \cdot b^2 + d_1 \cdot b + d_0$ , где  $b$  — основание системы счисления (в этой задаче оно равно  $\sqrt{-4}$ ). Таким образом могут получаться числа вида  $x + y\sqrt{-4}$ , где  $x$  и  $y$  — любые целые числа. Например:  $1030003_{\sqrt{-4}} = 1 \cdot (\sqrt{-4})^6 + 3 \cdot (\sqrt{-4})^4 + 3 \cdot (\sqrt{-4})^0 = -64 + 3 \cdot 16 + 3 = -13$ .

Несмотря на то, что операция взятия корня из отрицательного числа неоднозначна (как  $\sqrt{-4}$ , так и  $-\sqrt{-4}$  дают в квадрате  $-4$ ), в качестве основания системы счисления всегда берётся одно и то же число.

### Формат входных данных

В первой строке входного файла задано число  $A$ , во второй — число  $B$ . Их длины не превосходят 1 000 000. Числа даны в системе счисления с основанием  $\sqrt{-4}$  и не содержат ведущих нулей; число 0 обозначается одним нулём.

### Формат выходных данных

В единственной строке выходного файла должно быть выведено число  $A+B$ , записанное в системе счисления с основанием  $\sqrt{-4}$ . Выведенное число не должно содержать ведущих нулей; число 0 должно обозначаться одним нулём.

### Примеры

aplusb.in	aplusb.out
2	10301
3	
210	1030023
213	

### Задача В. Капча [2 секунды, 256 МиБ]

Володя собирается защитить свой чрезвычайно популярный сайт от хакерских атак. Для этого он изобрёл собственный полностью автоматизированный публичный тест Тьюринга для отличения компьютеров от людей (Completely Automated Public Turing test to tell Computers and Humans Apart, сокращённо CAPTCHA). Теперь каждый раз при заходе на сайт необходимо сообщить тесту, какой правильный многоугольник — правильный треугольник, квадрат или правильный пятиугольник — изображён на картинке Володиной капчи. Ваша задача — доказать Володе, что его тест можно обойти при помощи компьютера. Напишите программу, которая обходит его замечательную капчу.

Последовательность генерации одного теста следующая:

1. Случайным образом из отрезка  $[150, 250]$  выбираются целые числа — высота и ширина картинки  $n$  и  $m$ .
2. Случайным образом выбирается  $k$  от 3 до 5 — количество вершин правильного многоугольника.
3. Случайным образом выбирается длина стороны, центр многоугольника и угол поворота таким образом, что многоугольник целиком содержится в целевом прямоугольнике  $m \times n$ , а длина стороны не меньше 50. Все числа, участвующие в процессе — вещественные.
4. Производится растеризация. Пиксель дискретной картинки  $m \times n$  заполняется, если сторона многоугольника пересекает его (то есть если расстояние между квадратом этого пикселя и границей многоугольника не превосходит  $eps = 10^{-7}$ ).

#### Формат входных данных

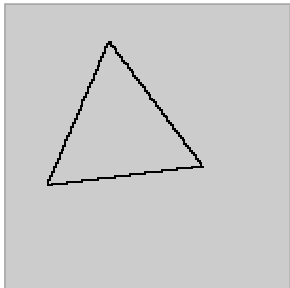
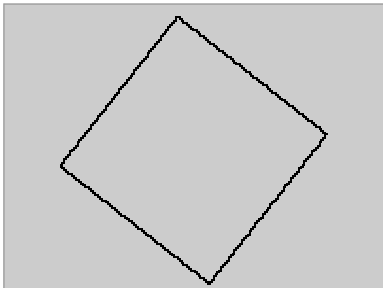
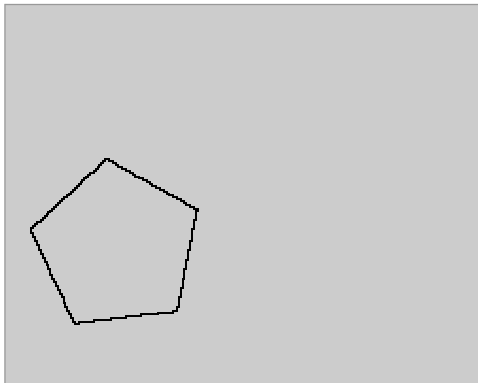
Первая строка содержит два целых числа  $n$  и  $m$  — высоту и ширину картинки ( $150 \leq n, m \leq 250$ ).

Следующие  $n$  строк содержат по  $m$  символов («#» означает, что пиксель закрашен, «.» — не закрашен).

#### Формат выходных данных

В первой строке выведите единственное слово в зависимости от того, что изображено на картинке — «TRIANGLE» для правильного треугольника, «SQUARE» для квадрата или «PENTAGON» для пятиугольника (без кавычек).

#### Примеры

Пример 1	Пример 2	Пример 3
$n = 150, m = 150$	$n = 150, m = 200$	$n = 200, m = 250$
		
TRIANGLE	SQUARE	PENTAGON

### **Пояснения к примерам**

Выше представлены лишь схематичные изображения примеров. Полные версии примеров доступны по следующей ссылке: <https://192.168.2.254/AS/day02/captcha/>.

### **Замечание**

Входной файл: `captcha.in`

Выходной файл: `captcha.out`

## Задача С. Электрическая цепь [2 секунды, 256 МиБ]

Идеальная девушка — девушка, сопротивлением которой при данных условиях можно пренебречь

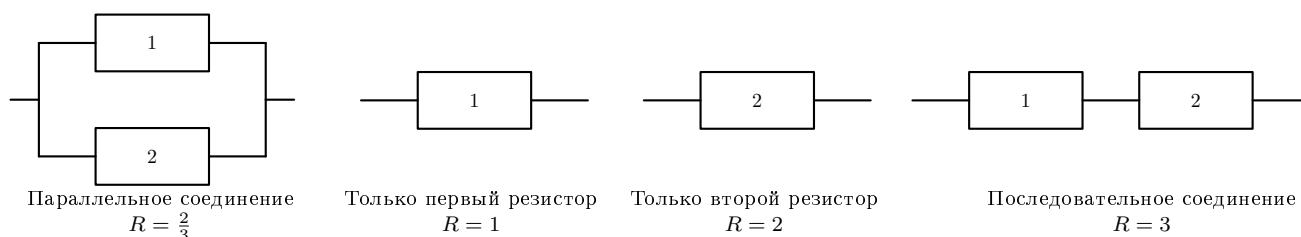
физический фольклор

У Васи есть  $N$  резисторов и неограниченное количество проводов. Он хочет собрать электрическую цепь с сопротивлением, как можно более близким к  $R$ . Вася умеет соединять две цепи параллельно или последовательно. Сопротивлением проводов можно пренебречь.

Электрические цепи и их сопротивления определяются следующим образом:

- Один резистор является электрической цепью; сопротивление такой цепи равно сопротивлению этого резистора.
- Если  $A$  и  $B$  — две электрические цепи, а  $R_A$  и  $R_B$  — их сопротивления, то, соединив  $A$  и  $B$  последовательно, мы получим новую электрическую цепь  $C$ , а её сопротивление будет равно  $R_C = R_A + R_B$ .
- Если  $A$  и  $B$  — две электрические цепи, а  $R_A$  и  $R_B$  — их сопротивления, то, соединив  $A$  и  $B$  параллельно, мы получим новую электрическую цепь  $C$ , а её сопротивление будет удовлетворять соотношению  $\frac{1}{R_C} = \frac{1}{R_A} + \frac{1}{R_B}$ .

Например, если у Васи есть два резистора, 1 Ом и 2 Ома, то он может собрать цепи с сопротивлениями  $\frac{2}{3}$ , 1, 2 и 3 Ома соответственно:



Отметим, что в цепи должен быть хотя бы один резистор.

### Формат входных данных

В первой строке входного файла заданы через пробел три числа  $N$ ,  $a$  и  $b$  ( $1 \leq N \leq 5$ ,  $R = \frac{a}{b}$ ,  $1 \leq b \leq 100$ ,  $0 \leq R \leq R_1 + R_2 + \dots + R_N$ ,  $a$  и  $b$  взаимно просты). Вторая строка содержит  $N$  целых положительных чисел  $R_i$  — сопротивления резисторов ( $1 \leq R_i \leq 100$ ).

### Формат выходных данных

Выведите  $|R - R_{opt}|$  — дробь в сокращённом виде (числитель и знаменатель дроби — взаимно простые неотрицательные целые числа). Здесь  $R$  — заданное во входном файле сопротивление, а  $R_{opt}$  — сопротивление цепи, которую может собрать Вася. Гарантируется, что числитель и знаменатель дроби не превосходят  $2^{63} - 1$ .

### Примеры

circuit.in	circuit.out
2 2 3 1 2	0/1
2 3 2 1 2	1/2

### Задача D. Очень важная формула [2 секунды, 256 МиБ]

Будучи студентом, двоечник Петя однажды уснул на лекции, на которой был доклад про очень важную формулу, которая могла бы дать ответы на все его вопросы:

$$C_n^0 - C_n^1 + C_n^2 - C_n^3 + \dots + (-1)^n C_n^n = 0.$$

Проснувшись, он с ужасом обнаружил, что проспал почти весь доклад! Не доверяя лектору, Петя решил проверить эту формулу на практике. Он выбрал число  $n$  и уже посчитал сумму первого  $k+1$  слагаемого, то есть  $C_n^0 - C_n^1 + \dots + (-1)^k C_n^k$ . Чтобы быть уверенным, что он ещё не успел ошибиться, он попросил вас проверить его вычисления по простому модулю  $m$ .

Напомним, что *биномиальный коэффициент*  $C_n^k$  — это количество способов выбрать подмножество из  $k$  различных элементов в множестве из  $n$  элементов. Известная формула для вычисления биномиальных коэффициентов в явном виде такова:

$$C_n^k = \frac{n!}{(n-k)! \cdot k!},$$

где  $x! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot x$  — факториал числа  $x$ . По определению  $0! = 1$ . Другая известная формула позволяет выразить биномиальные коэффициенты друг через друга рекуррентно:

$$C_n^k = C_{n-1}^{k-1} + C_{n-1}^k \text{ для } 0 < k < n,$$

$$C_n^0 = C_n^n = 1.$$

#### Формат входных данных

В первой и единственной строке ввода даны три целых числа  $n$ ,  $k$  и  $m$  ( $1 \leq k \leq n \leq 10^9$ ,  $m \leq 10^6 + 3$ ). Гарантируется, что число  $m$  — простое.

#### Формат выходных данных

Выведите единственное число  $r$  — остаток от деления искомой суммы на  $m$ . Это число должно удовлетворять соотношениям  $0 \leq r < m$ .

#### Примеры

cnk.in	cnk.out
2 1 3	2
5 2 13	6

#### Пояснения к примерам

В первом примере  $C_2^0 - C_2^1 = 1 - 2 = -1 \equiv 2 \pmod{3}$ .

Во втором примере  $C_5^0 - C_5^1 + C_5^2 = 1 - 5 + 10 = 6 \equiv 6 \pmod{13}$ .

## Задача Е. Бубновый джокер [2 секунды, 256 МиБ]

All your answers will be questioned

---

«Interstate 60»

У Арсения есть карта. И не одна, а целых 54. Не хватает только бубнового джокера. (Как же, вы не знаете, зачем нужен бубновый джокер?) Зато у него есть прямоугольный лист бумаги из тетрадки, из которого можно этого джокера вырезать.

Каждая клетка листа раскрашена в один из 26 цветов, а джокер должен представлять собой ромб, состоящий из клеток одного цвета. (Не обязательно красного; чёрный, или, скажем, синий бубновый джокер никого не смутит).

В данной задаче *ромбом* с центром в клетке  $(r_0, c_0)$  ( $r$  — номер строки,  $c$  — номер столбца) радиуса  $R$  называется множество клеток  $(r_i, c_i)$ , удовлетворяющих неравенству  $|r_i - r_0| + |c_i - c_0| \leq R$ .

Конечно, бóльший джокер полезнее в игре, так что Арсений хочет вырезать из бумаги самый большой ромб, состоящий из клеток одинакового цвета. Напишите программу, которая ему в этом поможет.

### Формат входных данных

В первой строке входного файла заданы через пробел два числа  $m$  и  $n$  ( $1 \leq m, n \leq 500$ ) — размеры прямоугольника (в клетках). Каждая из  $m$  последующих строк содержит по  $n$  прописных латинских букв, каждая латинская буква соответствует определённому цвету. Вторая строка во входном файле соответствует первой строке прямоугольника,  $(m+1)$ -я строка соответствует  $m$ -й строке прямоугольника.

### Формат выходных данных

Выведите в выходной файл три числа  $r_0$ ,  $c_0$  и  $R$  через пробел — номер строки и номер столбца центра, а также радиус ромба наибольшего размера. Если таких ромбов несколько, выведите ромб с наименьшим номером строки. В случае неоднозначности, выведите ромб с наименьшим номером столбца.

Выведенные числа также должны удовлетворять неравенствам:

$$1 + R \leq r_0 \leq m - R, 1 + R \leq c_0 \leq n - R,$$

то есть ромб для джокера должен целиком лежать в прямоугольнике.

### Пример

diamond.in	diamond.out
4 5 ABAAA AAAAA AAAAA AAAAA	2 3 1

## Задача F. Лекция [2 секунды, 256 МиБ]

Образование — это то, что остаётся, когда забываешь всё, чему учился в школе.

---

Эйнштейн

Константин Игоревич читает завтра лекцию по алгебре. Сегодня же ему надо решить, какие из теорем, которые он будет рассказывать, следует доказать. С одной стороны, лекция не может длиться более  $T$  минут, так что на доказательство вообще всех теорем времени, скорее всего, не хватит. С другой стороны, хочется, чтобы лекция оказалась для студентов как можно полезнее.

*Полезность лекции* определяется как сумма полезностей всех доказанных теорем, в то время как *полезность теоремы* — какое-то заранее известное для каждой теоремы число. Теоремы делятся на три класса по сложности, и их доказательство занимает одну, две или три минуты в зависимости от класса.

Конечно же, прилежные студенты заранее выучили все определения и формулировки, так что всё время лекции можно потратить на доказательство теорем.

### Формат входных данных

В первой строке входного файла заданы через пробел два числа  $N$  и  $T$  ( $1 \leq N \leq 100\,000$ ,  $1 \leq T \leq 300\,000$ ) — количество теорем и продолжительность лекции (в минутах). Далее идут  $N$  строк, в каждой два числа  $t_i$  и  $a_i$  ( $1 \leq t_i \leq 3$ ,  $0 \leq a_i \leq 10^9$ ), обозначающих класс сложности  $i$ -й теоремы (то есть, время в минутах, требуемое на её доказательство) и её полезность.

### Формат выходных данных

Выведите одно число — максимальную полезность завтрашней лекции Константина Игоревича.

### Примеры

lecture.in	lecture.out
3 3 1 3 2 10 2 14	17
3 4 2 5 2 10 3 14	15



## Задача G. Солдаты-самоубийцы [2 секунды, 256 МиБ]

Нет человека — нет проблемы

---

решение задачи Иосифа Виссарионовича

В ходе войны отряд из  $m \cdot n$  солдат был окружён противником. Чтобы избежать плена, солдаты решили покончить жизнь самоубийством.

Они выстроились в  $m$  рядов по  $n$  солдат в каждом ряду. Расстояние между рядами солдат оказалось равным расстоянию между солдатами в отдельном ряду и составило ровно один метр. Каждый солдат повернулся в одном из восьми направлений (восток, северо-восток, север, северо-запад, запад, юго-запад, юг, юго-восток) по направлению к своему дому. Солдаты могли быть призваны в армию из разных городов. Каждые десять секунд солдаты производят залп, причём каждый стреляет в том направлении, куда он повернут. Каждая пуля летит по прямой и убивает первого стоящего на её пути солдата, или улетает за пределы видимости, если солдат на её пути не оказалось. Предполагается, что все солдаты разного роста, поэтому две пули не могут столкнуться. Выжившие после залпа солдаты поворачиваются на 45 градусов против часовой стрелки. Например, солдат, выстреливший на восток, выстрелит в следующий раз на северо-восток. Известно, что боезапас каждого солдата равен  $100 \cdot m \cdot n$  патронов. Ваша задача состоит в том, чтобы определить количество выживших солдат.

### Формат входных данных

В первой строке входного файла заданы через пробел два числа  $m$  и  $n$  ( $1 \leq m, n \leq 200$ ). В  $m$  последующих строках заданы начальные направления солдат. Каждая строка состоит ровно из  $n$  десятичных цифр без пробелов. Направление задаётся десятичной цифрой  $a_{ij}$  ( $0 \leq a_{ij} \leq 7$ ); чтобы получить необходимое направление, надо повернуть солдата, смотрящего на восток, на  $45 \times a_{ij}$  градусов против часовой стрелки. Первый ряд самый северный, последний — самый южный. В каждой строке первая десятичная цифра задаёт направление для самого западного солдата, последняя — для самого восточного.

### Формат выходных данных

Выведите в выходной файл одно число — количество выживших солдат.

### Примеры

soldiers.in	soldiers.out
3 4 0000 4444 0000	2
2 2 06 24	0
2 1 4 4	1

## Задача Н. Квадратура круга [2 секунды, 256 МиБ]

— Извини, Теодор, но это ты очень странно рассуждаешь. Бессмыслица — искать решение, если оно и так есть. Речь идёт о том, как поступать с задачей, которая решения не имеет. Это глубоко принципиальный вопрос, который, как я вижу, тебе, прикладнику, к сожалению, не доступен.

---

Аркадий и Борис Стругацкие, «Понедельник начинается в субботу»

*Квадратура круга* — задача, заключающаяся в нахождении построения с помощью циркуля и линейки квадрата, равновеликого данному кругу (то есть имеющего ту же площадь, что и круг). Наряду с трисекцией угла и удвоением куба, эта задача является одной из самых известных неразрешимых задач на построение с помощью циркуля и линейки. Однако, задача о квадратуре круга становится разрешимой, если расширить средства построения, а также если искать не точное, а приближённое решение.

В этой задаче требуется по кругу, заданному координатами центра и радиусом, построить квадрат, площадь которого отличается от площади этого круга не более чем на  $10^{-6}$ . В качестве средства предлагается использовать компьютер и один из доступных языков программирования.

Напомним, что площадь квадрата со стороной  $a$  равна  $a^2$ , а площадь круга радиуса  $r$  равна  $\pi \cdot r^2$ , где  $\pi \approx 3.1415926535897932384626433832795\dots$  — это половина длины окружности единичного радиуса.

### Формат входных данных

В единственной строке входного файла заданы через пробел три целых числа  $x$ ,  $y$  и  $r$  ( $|x|, |y| \leq 100, 1 \leq r \leq 100$ ) — координаты центра круга и его радиус.

### Формат выходных данных

Выведите в выходной файл четыре строки. Каждая строка должна содержать два числа через пробел — координаты одной из вершин квадрата. Найденный квадрат должен иметь стороны, параллельные осям координат, и площадь, равную площади данного круга, а его центр должен совпадать с центром круга. В первой строке выведите координаты левой нижней вершины квадрата, во второй — левой верхней, в третьей — правой верхней и в четвёртой — правой нижней вершины.

Выводите вещественные числа как можно более точно! Допускается экспоненциальная форма вывода. При проверке ответов **все** проверки на равенство — сравнение координат точек и площадей квадратов — производятся с точностью до  $10^{-6}$ .

### Пример

squaring.in	squaring.out
2 3 5	-2.431134627264 -1.431134627264 -2.431134627264 7.431134627264 6.431134627264 7.431134627264 6.431134627264 -1.431134627264

## Задача I. Расписание [2 секунды, 256 МиБ]

Раз Писание, два Писание. . .

---

из описи имущества церкви

Арсений приехал в очередной раз в университет и, подойдя к расписанию, узнал, что сегодня читается ровно  $N$  лекций. Также он узнал времена начала и конца каждой лекции и фамилии лекторов. Теперь он в раздумье — куда же следует пойти. Поскольку у Арсения с собой всего одна тетрадка, он не может посетить более  $K$  различных лекций, иначе ему просто негде будет вести конспект. Лекции проводятся в различных аудиториях, так что в каждый момент времени Арсений может присутствовать не более чем на одной лекции. Но он может уходить с лекции в любой момент, даже не дождавшись конца, и приходить после начала.

За каждую минуту, проведённую на лекции, Арсений получает 1 пункт знаний. Естественно, он хочет максимизировать суммарные знания, полученные за весь день. Помогите Арсению понять, чего он сможет достичь.

Арсений очень хорошо ориентируется на своём факультете, так что временем, затрачиваемым на переходы между лекционными аудиториями, можно пренебречь.

### Формат входных данных

В первой строке входного файла заданы через пробел два числа  $N$  и  $K$  ( $1 \leq N \leq 100\,000$ ,  $0 \leq K \leq 100$ ). Далее идут  $N$  строк, каждая из которых содержит два числа  $s_i$  и  $e_i$  ( $0 \leq s_i \leq e_i \leq 10^9$ ), задающих время начала и конца  $i$ -й лекции соответственно. Время задаётся в минутах с момента приезда Арсения. Все числа во входном файле целые.

### Формат выходных данных

Выведите одно число — максимальное число пунктов знаний, которые Арсений сможет получить за этот день (день у Арсения заканчивается не раньше, чем закончится последняя лекция).

### Примеры

timetable.in	timetable.out
3 2 1 2 3 6 5 8	5
3 2 1 3 4 6 5 7	4

## Задача J. Изгрызенные книги [2 секунды, 256 МиБ]

— Тысяча червей!

из диалога преферансистов

Санкт-Петербургский Государственный Университет славится, в частности, своей библиотекой. Однако, другой известный университет из зависти запустил в библиотеку СПбГУ книжных червей. Теперь главному библиотекарю (его, кстати говоря, зовут Вася) нужно срочно определить величину ущерба.

Все  $N$  книг в библиотеке хранятся на одной очень длинной полке. Посмотрев на корешок, Вася может определить номер книги, не трогая её руками. Книги пронумерованы слева направо, начиная с единицы. Ни одна книга не перевёрнута вверх тормашками.

Вася обнаружил в библиотеке  $M$  червей. Он определил, где каждый червь начал и где закончил свой путь. Все черви двигались прямолинейно слева направо или справа налево. Чтобы правильно посчитать ущерб, нанесённый библиотеке, Вася хочет написать программу, вычисляющую, сколько страниц изгрызло ровно  $k$  червей. Помогите ему это сделать.



### Формат входных данных

В первой строке входного файла заданы через пробел два числа  $N$  и  $M$  ( $1 \leq N \leq 10\,000$ ,  $1 \leq M \leq 100\,000$ ). Вторая строка содержит  $N$  положительных целых чисел  $p_i$  — количество страниц в  $i$ -й книге ( $p_i \leq 10\,000$ ). В следующих  $M$  строках содержатся описания путей. Описание пути состоит из четырёх положительных целых чисел — номер книги и номер страницы начала пути червя, а также номер книги и номер страницы окончания пути червя.

### Формат выходных данных

Выходной файл должен содержать  $(M + 1)$  строк. В  $k$ -й строке выведите количество страниц, которые изгрызло ровно  $(k - 1)$  червей.

### Пример

worms.in	worms.out
3 2	0
1 1 1	2
1 1 3 1	1
2 1 2 1	