

Задача 1. Ролевая игра

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

Вася готовит инвентарь для ролевой игры. В игре должны принять участие n игроков, каждый из которых будет изображать персонажа фантастического мира. В процессе игры каждый персонаж будет обладать некоторым *уровнем* x , который представляет собой целое число от 1 до m .

Для обозначения уровня планируется использовать специальные значки двух цветов. Белый значок обозначает один уровень, а красный значок — k уровней. Игрок, изображающий персонажа с уровнем x , должен иметь a белых значков и b красных значков, чтобы сумма $(a + bk)$ была равна x . При этом персонажу не разрешается иметь более чем $(k - 1)$ белых значков.

Значки для игры готовятся заранее, однако уровни персонажей заранее неизвестны. Для успешного проведения игры всем персонажам необходимо выдать соответствующее их уровням количество значков. Возникает вопрос: какое минимальное суммарное количество значков необходимо подготовить для успешного проведения игры при любых уровнях участвующих персонажей.

Требуется написать программу, которая по заданным числам n , m и k вычисляет минимальное количество значков, которое необходимо подготовить для успешного проведения игры.

Формат входного файла

Входной файл содержит расположенные в одной строке три целых числа: n , m и k ($1 \leq n \leq 10^4$, $1 \leq m \leq 10^5$, $1 \leq k \leq 10^5$).

Формат выходного файла

В выходном файле должно содержаться одно целое число — минимальное количество значков, которое требуется подготовить.

Пример входных и выходных данных

<code>game.in</code>	<code>game.out</code>
3 4 2	9

Пояснения к примеру

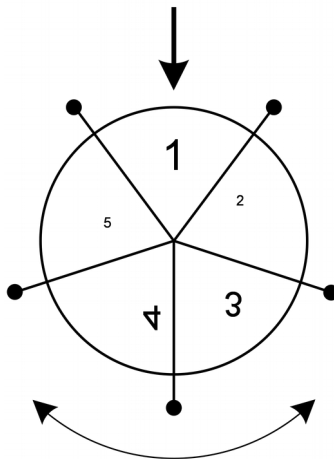
В приведенном примере необходимо подготовить 6 красных и 3 белых значка.

Задача 2. «Колесо Фортуны»

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

Развлекательный телеканал транслирует шоу «Колесо Фортуны». В процессе игры участники шоу крутят большое колесо, разделенное на сектора. В каждом секторе этого колеса записано число. После того как колесо останавливается, специальная стрелка указывает на один из секторов. Число в этом секторе определяет выигрыш игрока.

Юный участник шоу заметил, что колесо в процессе вращения замедляется из-за того, что стрелка задевает за выступы на колесе, находящиеся между секторами. Если колесо вращается с угловой скоростью v градусов в секунду, и стрелка, переходя из сектора X к следующему сектору, задевает за очередной выступ, то текущая угловая скорость движения колеса уменьшается на k градусов в секунду. При этом если $v \leq k$, то колесо не может преодолеть препятствие и останавливается. Стрелка в этом случае будет указывать на сектор X .



Юный участник шоу собирается вращать колесо. Зная порядок секторов на колесе, он хочет заставить колесо вращаться с такой начальной скоростью, чтобы после остановки колеса стрелка указала на как можно большее число. Колесо можно вращать в любом направлении и придавать ему начальную угловую скорость от a до b градусов в секунду.

Требуется написать программу, которая по заданному расположению чисел в секторах, минимальной и максимальной начальной угловой скорости вращения колеса и величине замедления колеса при переходе через границу секторов вычисляет максимальный выигрыш.

Формат входного файла

Первая строка входного файла содержит целое число n — количество секторов колеса ($3 \leq n \leq 100$).

Вторая строка входного файла содержит n положительных целых чисел, каждое из которых не превышает 1000 — числа, записанные в секторах колеса. Числа приведены в порядке следования секторов по часовой стрелке. Изначально стрелка указывает на первое число.

Третья строка содержит три целых числа: a , b и k ($1 \leq a \leq b \leq 10^9$, $1 \leq k \leq 10^9$).

Формат выходного файла

В выходном файле должно содержаться одно целое число — максимальный выигрыш.

Примеры входных и выходных данных

wheel.in	wheel.out
5 1 2 3 4 5 3 5 2	5
5 1 2 3 4 5 15 15 2	4
5 5 4 3 2 1 2 5 2	5

Пояснения к примерам

В первом примере возможны следующие варианты: можно придать начальную скорость колесу равную 3 или 4, что приведет к тому, что стрелка преодолеет одну границу между секторами, или придать начальную скорость равную 5, что позволит стрелке преодолеть 2 границы между секторами. В первом варианте, если закрутить колесо в одну сторону, то выигрыш получится равным 2, а если закрутить его в противоположную сторону, то — 5. Во втором варианте, если закрутить колесо в одну сторону, то выигрыш будет равным 3, а если в другую сторону, то — 4.

Во втором примере возможна только одна начальная скорость вращения колеса – 15 градусов в секунду. В этом случае при вращении колеса стрелка преодолеет семь границ между секторами. Тогда если его закрутить в одном направлении, то выигрыш составит 4, а если в противоположном направлении, то — 3.

Наконец, в третьем примере оптимальная начальная скорость вращения колеса равна 2 градусам в секунду. В этом случае стрелка вообще не сможет преодолеть границу между секторами, и выигрыш будет равен 5.

Примечание

Правильные решения для тестов, в которых $1 \leq a \leq b \leq 1000$, будут оцениваться из 50 баллов.

Задача 3. Форматирование текста

Имя входного файла:	<code>format.in</code>
Имя выходного файла:	<code>format.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Многие системы форматирования текста, например $\text{T}_\text{E}\text{X}$ или Wiki, используют для разбиения текста на абзацы пустые строки. Текст представляет собой последовательность слов, разделенных пробелами, символами перевода строк и следующими знаками препинания: «,», «.», «?», «!», «-», «:» и «'» (ASCII коды 44, 46, 63, 33, 45, 58, 39). Каждое слово в тексте состоит из заглавных и прописных букв латинского алфавита и цифр. Текст может состоять из нескольких абзацев. В этом случае соседние абзацы разделяются одной или несколькими пустыми строками. Перед первым абзацем и после последнего абзаца также могут идти одна или несколько пустых строк.

Дальнейшее использование исходного текста предполагает его форматирование, которое осуществляется следующим образом. Каждый абзац должен быть разбит на строки, каждая из которых имеет длину не больше w . Первая строка каждого абзаца должна начинаться с отступа, состоящего из b пробелов. Слова внутри одной строки должны быть разделены ровно одним пробелом. Если после слова идет один или несколько знаков препинания, они должны следовать сразу после слова без дополнительных пробелов. Если очередное слово вместе со следующими за ним знаками препинания помещается на текущую строку, оно размещается на текущей строке. В противном случае, с этого слова начинается новая строка. В отформатированном тексте абзацы не должны разделяться пустыми строками. В конце строк не должно быть пробелов.

Требуется написать программу, которая по заданным числам w и b и заданному тексту выводит текст, отформатированный описанным выше образом.

Формат входного файла

Первая строка входного файла содержит два целых числа: w и b ($5 \leq w \leq 100$, $1 \leq b \leq 8$, $b < w$).

Затем следует одна или более строк, содержащих заданный текст. Длина слова в тексте вместе со следующими за ним знаками препинания не превышает w , а длина первого слова любого абзаца вместе со следующими за ним знаками препинания не превышает $(w - b)$.

Размер входного файла не превышает 100 Кбайт. Длина каждой строки во входном файле не превышает 250.

Формат выходного файла

Выходной файл должен содержать заданный текст, отформатированный в соответствии с приведенными в условии задачи правилами.

Примеры входных и выходных данных

format.in
20 4 Yesterday, All my troubles seemed so far away, Now it looks as though they're here to stay, Oh, I believe in yesterday. Suddenly, I'm not half the man I used to be, There's a shadow hanging over me, Oh, yesterday came suddenly...
format.out
Yesterday, All my troubles seemed so far away, Now it looks as though they' re here to stay, Oh, I believe in yesterday. Suddenly, I' m not half the man I used to be, There' s a shadow hanging over me, Oh, yesterday came suddenly...

Примечание

Правильные решения для тестов, в которых заданный текст состоит из одного абзаца и входной файл не содержит пустых строк, будут оцениваться из 30 баллов.

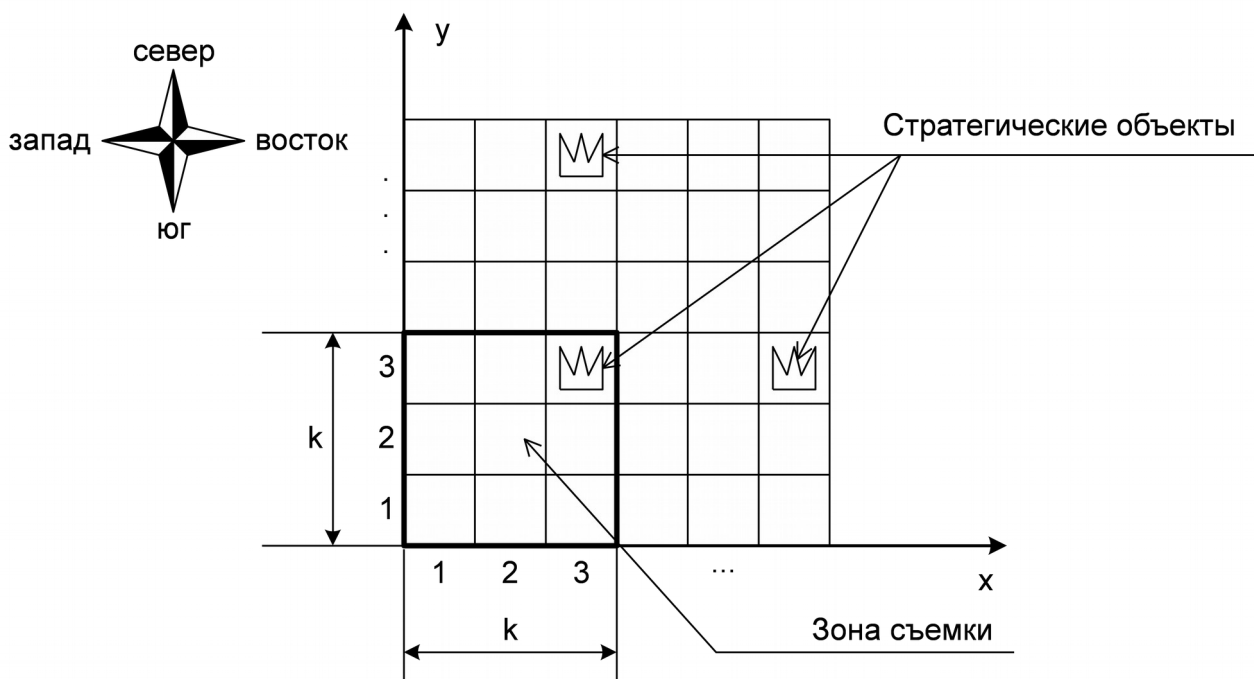
Правильные решения для тестов, в которых соседние слова разделены ровно одним пробелом и все знаки препинания следуют сразу за словами и не отделены от них пробелами или символами перевода строк, будут оцениваться из 30 баллов.

Задача 4. Космические исследования

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

Отделу космических исследований поступило задание сфотографировать из космоса n объектов в заданной области. Область имеет форму квадрата размером 50×50 километров. Если разделить ее на квадраты размером 1×1 километр, то интересные отдел объекты окажутся в центрах некоторых единичных квадратов.

Введем систему координат, направив ось OX с запада на восток и ось OY с юга на север. Тогда каждому единичному квадрату будут сопоставлены координаты в диапазоне от 1 до 50, как показано на рисунке ниже.



Для космической съемки используется специальный фотоаппарат высокого разрешения, установленный на космическом спутнике. Фотоаппарат может делать снимки квадратных участков земной поверхности размером $k \times k$ километров. Исходно аппарат наведен на юго-западный угол заданной области, то есть, если сделать снимок, на нем будут видны единичные квадраты с координатами x и y от 1 до k километров.

С помощью специальных двигателей можно изменять орбиту спутника, что приводит к изменению участка съемки. За один день орбиту спутника можно изменить таким образом, что участок съемки сместится либо на один километр на запад, либо на один километр на восток, либо на один километр на север. Переместить участок съемки на юг невозможно. Непосредственно между перемещениями спутника можно сделать снимок, временем съемки можно пренебречь.

Руководство отдела заинтересовалось вопросом: за какое минимальное количество дней можно сделать снимки всех объектов заданной области.

Требуется написать программу, которая по заданному расположению объектов и размеру снимка k определит минимальное время, за которое можно сделать снимки всех объектов заданной области.

Формат входного файла

Первая строка входного файла содержит два целых числа: n и k ($1 \leq n \leq 1000$, $1 \leq k \leq 5$).

Следующие n строк содержат по два целых числа: x_i и y_i — координаты объектов в заданной области ($1 \leq x_i, y_i \leq 50$).

Формат выходного файла

В выходном файле должно содержаться одно целое число: минимальное количество дней, которое требуется для получения снимков всех объектов в заданной области.

Примеры входных и выходных данных

space.in	space.out
4 1 1 1 10 2 1 3 10 4	30
4 2 1 1 10 2 1 3 10 4	10
1 1 1 1	0
3 3 3 3 3 6 6 3	7

Пояснения к примерам

В первом примере возможна следующая последовательность действий: сделать снимок, 9 раз сместиться на восток, сместиться на север, сделать снимок, 9 раз сместиться на запад, сместиться на север, сделать снимок, 9 раз сместиться на восток, сместиться на север, сделать снимок. Всего требуется 30 перемещений участка съемки.

Во втором примере объекты расположены там же, но размер снимка больше, поэтому можно действовать так: сделать снимок, сместиться на север, сделать снимок, 8 раз сместиться на восток, сделать снимок, сместиться на север, сделать снимок. Всего требуется лишь 10 перемещений участка съемки.

В третьем примере перемещать участок съемки не требуется, можно просто сделать снимок.

Четвертый пример соответствует приведенному выше рисунку.

Примечание

Правильные решения для тестов, в которых $k = 1$, будут оцениваться из 30 баллов.

Правильные решения для тестов, в которых $k > 1$ и $1 < n \leq 15$, будут оцениваться из 30 баллов.