

Дамка

Ограничение времени 2 секунды

Ограничение памяти 64Mb

Ввод стандартный ввод или input.txt

Вывод стандартный вывод или output.txt

Как известно, для игры в шашки можно использовать шахматную доску, в которой горизонтальные строки обозначаются цифрами от 1 до 8, считая снизу вверх, а вертикальные столбцы — буквами латинского алфавита: a, b, c, d, e, f, g, h.

Для начинающих играть в шашки часто задают такую задачу: размещают белую шашку на доске и просят определить, сможет ли эта шашка попасть в заданную клетку, делая ходы по правилам. По правилам белая шашка ходит по чёрным полям по диагонали вверх. Если шашка достигает последней горизонтали, то она превращается в дамку и может ходить по диагонали в любом направлении на любое число клеток.

Требуется написать программу, определяющую возможность перемещения белой шашки из одной заданной клетки в другую заданную клетку за минимальное число ходов.

Формат ввода

В единственной строке входных данных записаны в шахматной нотации: клетка, где стоит шашка, затем через пробел клетка, куда шашка должна попасть. Начальная и конечная клетки не совпадают.

Формат вывода

В единственную строку выходных данных нужно вывести минимальное количество ходов, за которое шашка или дамка может попасть из начальной клетки в конечную, и число -1 в противном случае.

Пример 1

Ввод **Вывод**

a1 b2 1

Пример 2

Ввод **Вывод**

b2 a1 7

Пример 3

Ввод **Вывод**

a1 h7 -1

Пример 4

Ввод Вывод

a1 h8 7

Примечания

Тест 1: Для выполнения указанного перемещения шашка должна сделать один ход вперед и вправо

Тест 2: Для выполнения указанного перемещения шашка должна сделать 6 ходов вправо вверх и после превращения в дамку вернуться обратно

Тест 3: a1 и h7 — клетки разного цвета

Тест 4: Требуется 7 ходов вправо вверх

Как стать призером

Ограничение времени 2 секунды

Ограничение памяти 64Mb

Ввод стандартный ввод или input.txt

Выход стандартный вывод или output.txt

К тренеру на занятия по подготовке к олимпиаде ходит N школьников. Для каждого из школьников заданы два параметра: начальный условный опыт A_i и условный интеллект B_i .

Каждое занятие устроено так: тренер подходит к какому-нибудь школьнику и обсуждает с ним возникшие вопросы и проблемы. В результате такого обсуждения условный опыт этого школьника возрастает на B_i (то есть чем выше условный интеллект школьника, тем больше этот школьник может взять из общения с тренером).

За все время подготовки к олимпиаде тренер может подойти ко всем школьникам суммарно не более C раз (он может подходить к разным школьникам, может несколько раз подходить к одному и тому же школьнику). Для того чтобы школьник стал призером олимпиады, к началу олимпиады его условный опыт должен быть не меньше, чем K .

Напишите программу, которая вычислит максимальное количество призеров олимпиады, которое сможет подготовить тренер.

Формат ввода

Сначала вводятся натуральные числа N , C , K , задающие количество школьников, количество подходов, которые может сделать учитель, и условный опыт, необходимый, чтобы стать призером олимпиады, соответственно ($1 \leq N \leq 10^6$, $1 \leq C \leq 10^9$, $1 \leq K \leq 10^9$). Далее идет N пар целых неотрицательных чисел A_i , B_i , задающих начальный условный опыт и условный интеллект каждого школьника. Каждое из чисел A_i и B_i не превышает 10^9 .

Формат вывода

Выведите одно число — наибольшее количество призеров олимпиады, которое успеет подготовить тренер.

Пример 1

Ввод **Выход**

```
3 5 6
1 1 2
2 1
4 2
```

Пример 2

Ввод **Выход**

Ввод Вывод

4 10 3
0 1
0 1 3
0 2
2 0

Примечания

Тест 1: Тренер подойдёт к школьнику №2 4 раза (интеллект станет $2+4*1=6$) и к школьнику №3 1 раз (интеллект станет $4+1*2=6$)

Тест 2: Тренер подойдёт к школьникам №1 и №2 по 3 раза (интеллект станет $3*1=3$) и к школьнику №3 2 раза (интеллект станет $2*2=4$). Оставшиеся подходы может потратить на любого из школьников, но нового призёра не получится.

Форматирование текста

Ограничение времени 2 секунды

Ограничение памяти 64Mb

Ввод стандартный ввод или input.txt

Выход стандартный вывод или output.txt

Многие системы форматирования текста, например TEX 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 ввода

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 вывода

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

Пример

Ввод

```
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...
```

Вывод

```
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...
```

Примечания

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.