

## Задача А. Ассоциативное мышление

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Профессор  $\hat{H}$  часто совершает важные открытия. Особенно хорошо у него это получается, когда он гуляет по бульвару. Вдоль всего бульвара установлены лавочки, и среди них нет ни одной пары одинаковых.

Профессор  $\hat{H}$  находит это обстоятельство совершенно замечательным, и вот почему. Он давно заметил, что время, необходимое для совершения очередного открытия, удивительным образом совпадает со временем, в течение которого он проходит мимо  $k$  лавочек. Поскольку у профессора  $\hat{H}$  чрезвычайно развито ассоциативное мышление, то ему достаточно посмотреть некоторое время на лавочку — и он уже не позабудет сделанное открытие.

После этого он отправляется гулять дальше, и, пройдя очередные  $k$  лавочек, запоминает очередное сделанное открытие. Каждый раз, когда он доходит до конца бульвара, он разворачивается и продолжает гулять и совершать открытия дальше. Однако профессор  $\hat{H}$  не может связывать с одной лавочкой более одного открытия. И если он, сделав очередное открытие, оказался возле лавочки, с которой уже связано другое открытие, он прекращает прогулку и отправляется домой, чтобы не забыть совершённое открытие.

Будем считать лавочки занумерованными от 1 до  $n$ . Первая лавочка расположена в самом начале бульвара, а лавочка  $n$  — в самом конце бульвара. Профессор начал свою прогулку от лавочки  $m$  и сначала отправился по направлению к лавочке  $n$ . Лавочку  $m$  в начале прогулки профессор не считает пройденной. Лавочки, возле которых он разворачивается (в начале и в конце бульвара), он считает пройденными единожды, а не дважды.

Ваша задача — определить, сколько открытий совершит профессор, прежде чем закончит прогулку.

### Формат входного файла

В первой строке содержатся целые числа  $n, m, k$  ( $2 \leq n \leq 5000, 1 \leq m \leq n, 1 \leq k \leq 5000$ ) — количество лавочек на бульваре, номер лавочки, возле которой начинается прогулка профессора, количество лавочек, которые должен миновать профессор для совершения очередного открытия.

### Формат выходного файла

В первой строке — количество открытий, которые совершит профессор, прежде чем завершит прогулку.

### Примеры

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

### Note

В первом примере профессор сначала минует четвёртую лавочку, затем пятую, вернется к четвёртой, а возле третьей совершит открытие. Затем он пройдет мимо второй, мимо первой, вновь мимо второй и вернется к третьей. Увы, она уже занята предыдущим открытием, и профессору придется пойти домой, совершив два открытия.

Во втором примере профессор сразу развернется и пойдет к началу бульвара. Он совершит открытия возле лавочек с номерами 3, 2, 5, 4, 1, и после этого вновь придет к лавочке 4. Таким образом, сделав 6 открытий, профессор отправится домой.

## Задача В. Точка, точка, запятая...

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Совершив очередное открытие, профессор  $\hat{H}$ , конечно же, пишет очередную статью в журнал. К сожалению, в последнее время на его любимой клавиатуре клавиша пробела работает не так хорошо, как хотелось бы: иногда её нажатие приводит к появлению в тексте нескольких пробелов подряд, иногда, напротив, пробел вовсе не печатается.

Профессор  $\hat{H}$  написал много статей, прежде чем обратил на это внимание. И теперь он хочет автоматизировать исправление опечаток, связанных с пробелами. В его статьях встречаются следующие знаки: точки (`.`), запятые (`,`), восклицательные знаки (`!`), вопросительные знаки (`?`), двоеточия (`:`), многоточия (`...`, три идущие подряд точки), точки с запятой (`;`), тире (`--`) и дефисы (`-`).

Согласно правилам, перед этими знаками, за исключением тире, пробел не ставится. После этих знаков, за исключением дефиса, пробел ставится.

В тексте статей, кроме перечисленных выше знаков, могут присутствовать строчные и заглавные латинские буквы. Переводы строк в тексте отсутствуют. Никакие два знака не могут идти подряд (в том числе через пробел(ы)). Текст начинается с буквы (перед которой не нужен пробел) и завершается точкой или многоточием (после которых пробел не ставится).

Профессору  $\hat{H}$  сейчас очень важно знать, какое количество символов будет содержать его статья, если в неё добавить недостающие и удалить лишние пробелы.

Ваша задача — написать программу, которая подсчитает количество символов в исправленной статье профессора  $\hat{H}$ .

### Формат входного файла

Входными данными является текст статьи профессора  $\hat{H}$ . Текст содержит не более 5000 символов, описанных в условии.

### Формат выходного файла

Целое число — количество символов в статье профессора после внесения исправлений.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
<code>My , name-- is H!I am:professor .Problem - B ;is very easy ...</code>	61
<code>Punctuation is a very important.</code>	32

## Задача С. В поисках непротиворечивости

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Недавно профессора  $\hat{H}$  пригласили поработать с группой археологов. Среди находок оказался папирус, в котором, помимо прочего, описывались состояния самых богатых горожан.

В папирусе встречались записи вида:

«Фенуку владеет имуществом стоимостью не менее, чем имущество Шушу, имущество Эбо, имущество Табита и еще раз имущество Табита, вместе взятых»

или

«Хорус владеет имуществом стоимостью более, чем имущество Табита, Фенуку, Джаха и Бакари, вместе взятых».

Профессор  $\hat{H}$  понял, что древние не умели обращаться с большими числами и заинтересовался, могут ли все утверждения папируса одновременно быть верными. Он построил математическую модель этих записей, состоящую из  $n$  неравенств (по числу записей в папирусе), используя в них вместо имён горожан переменные вида  $x_j$ , где  $j$  — целое положительное число, не превосходящее 100.

Таким образом, у него получились записи вида:

$$x_1 \geq x_2 + x_2 + x_3 + x_5,$$

$$x_6 > x_2 + x_1 + x_8 + x_{11}.$$

Ваша задача — найти набор чисел  $x_j$  ( $1 \leq x_j \leq 10^{16}$ ), который удовлетворяет системе неравенств. В случае, если решение не существует, следует вывести единственное число 0.

### Формат входного файла

В первой строке содержатся целые числа  $n$  и  $m$  ( $1 \leq n \leq 10, 1 \leq m \leq 100$ ) — количество неравенств и наибольший номер переменной, которая в них используется (нумерация переменных с единицы).

В каждой из последующих  $n$  строк содержится по одному неравенству, подобному описанному в условии. В левой части каждого неравенства записана единственная переменная, в правой части — сумма переменных. В неравенствах используются только знаки  $>$  и  $\geq$ . В правой части неравенства сумма состоит из не более чем десяти слагаемых.

Гарантируется, что если система имеет решение в целых положительных числах, то она имеет и решение в ограничениях ( $1 \leq x_j \leq 10^{16}$ ).

### Формат выходного файла

В каждой из  $m$  строк содержится возможное значение переменной с соответствующим номером, если система неравенств допускает решение.

В противном случае в единственной строке содержится единственное число 0.

## Примеры

input.txt	output.txt
2 11 x1 >= x2 + x2 + x3 + x5 x6 > x1 + x2 + x8 + x11	4 1 1 1 1 8 1 1 1 1 1
2 2 x1 > x2 x2 > x1	0
2 2 x1 >= x2 x2 >= x1	1 1
5 5 x2 >= x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 + x3 x1 >= x3 x3 > x5 + x4 x1 > x2 + x5 x4 > x5	42 40 4 2 1

## Задача D. Надпись на стене

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Ещё одной находкой археологов оказалась надпись на городской стене. Эта надпись содержит  $n$  символов.

Профессор  $\hat{H}$  предполагает, что надпись имела некое «охраняющее» значение и содержала в том числе многократно повторённое имя покровителя города. Также профессор абсолютно уверен в том, что имя покровителя состоит ровно из  $2^k$  символов. Наконец, он совершенно убеждён, что среди всех подстрок длиной  $2^k$  символов имя покровителя города — наиболее часто встречающаяся в надписи подстрока.

Ваша задача — по заданной надписи найти подстроку длины  $2^k$ , которая встречается в ней чаще всего. Если таких подстрок несколько, выберите среди них лексикографически минимальную (т.е. ту, которая окажется первой в словаре, если внести в него все найденные варианты).

### Формат входного файла

В первой строке содержатся целые числа  $n$  и  $k$  ( $1 \leq n \leq 10^5$ ,  $1 \leq 2^k \leq n$ ).

Во второй строке содержится строка из  $n$  строчных латинских букв.

### Формат выходного файла

В единственной строке — найденная подстрока.

### Примеры

<code>input.txt</code>	<code>output.txt</code>
35 2 sitametconsecteturacommodoconsequat	cons
26 0 zyxwvutsrqponmlkjihgfedcba	a

### Note

В первом примере существуют две подстроки длины 4, которые входят в строку дважды: это «cons» и «onse».

Для лексикографического сравнения необходимо найти первую позицию, в которой эти подстроки различаются. Та подстрока, в которой буква в этой позиции предшествует в алфавите букве в этой же позиции в другой подстроке, и будет меньше. В данном случае это позиция 1 (при нумерации с единицы), буква «с» предшествует в алфавите букве «о», так что ответом будет подстрока «cons».

## Задача Е. Весомые аргументы

Имя входного файла: `input.txt`  
Имя выходного файла: `output.txt`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Профессор  $\hat{H}$  успешно работает в самых разных областях науки. Недавно он получил приглашения на две конференции — по археологии и по криптографии. К сожалению, обе они проходят в одно и то же время в разных городах. Это значит, что профессору придется выбирать, куда же он поедет.

Профессор  $\hat{H}$  решил подойти к выбору очень тщательно. «Конференция по археологии проходит в очень красивом городе, и это плюс». — рассуждал он. — «Но на конференцию по криптографии планирует приехать профессор  $\hat{L}$  с докладом о новом методе построения шифров. Это тоже плюс. К тому же с ним можно обсудить мой новый результат, а это большой плюс. Даже два плюса».

По мере возникновения тех или иных доводов в пользу поездки на ту или иную конференцию профессор записывал количество плюсов, которые дают эти доводы. Правда, чтобы не запутаться, перед количеством плюсов, которые дает конференция по криптографии, он записывал минус.

Он решил, что если количество плюсов какой-либо конференции окажется хотя бы на  $p$  больше, чем количество плюсов другой конференции, то он поедет на неё. В противном же случае он запросит у организаторов обеих конференций дополнительную информацию.

Ваша задача — определить, какое решение примет профессор  $\hat{H}$ .

### Формат входного файла

В первой строке содержится целые числа  $n$  и  $p$  ( $1 \leq n \leq 1000, 1 \leq p \leq 1000$ ) — общее количество доводов, сформулированных профессором  $\hat{H}$ , и разница в количестве плюсов, которая позволит профессору сделать выбор.

Во второй строке содержится  $n$  положительных и отрицательных чисел — количество плюсов, которые дают конференции. Каждое из этих чисел не превосходит по абсолютной величине 1000.

### Формат выходного файла

Выведите единственное целое число:

- 0 — если профессор запросит дополнительную информацию,
- 1 — если профессор поедет на конференцию по археологии,
- 2 — если профессор поедет на конференцию по криптографии.

### Примеры

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