

## Задача А. Все слова

Имя входного файла: `words.in`  
Имя выходного файла: `words.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Вася придумал свой собственный алфавит, в котором  $N$  символов. Теперь он хочет составить с помощью этого алфавита все слова, состоящие ровно из  $K$  букв, причём ни одно слово не может начинаться с первой буквы алфавита и не может заканчиваться на последнюю букву алфавита. Помогите Васе определить, сколько он сможет составить таких слов.

### Формат входных данных

Входная строка содержит два числа, разделённых пробелом: число символов в алфавите  $N$  и длину слов  $K$  ( $0 < N, K \leq 10$ ).

### Формат выходных данных

Программа должна вывести единственное число — количество слов длиной  $K$  в алфавите мощностью  $N$ , таких что ни одно слово не начинается с первой буквы алфавита и не заканчивается на последнюю букву алфавита.

### Примеры

<code>words.in</code>	<code>words.out</code>
5 5	2000

## Задача В. Планетоход

Имя входного файла: `planet.in`  
Имя выходного файла: `planet.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

В конструкторском бюро проектируют планетоход для исследования поверхности планеты Марс. Исследования должны проводиться на прямоугольной области планеты без препятствий внутри неё. Эта область разделена на единичные квадраты и имеет размеры  $M \times N$ , где  $M$  — высота прямоугольника, а  $N$  — его ширина.

Планируется, что планетоход должен работать по следующей программе. Вначале он садится в северо-западном углу заданной области в направлении на восток. После этого планетоход начинает обход и исследование выбранной области, двигаясь по спирали по часовой стрелке. При этом спираль постепенно "закручивается" вовнутрь, захватывая постепенно все клетки прямоугольника. Исследование заканчивается, когда пройдены все клетки (после очередного поворота планетохода).

Требуется написать программу, которая для заданных  $M$  и  $N$  определяет количество поворотов, которые должен выполнить планетоход в процессе исследования области.

### Формат входных данных

В единственной входной строке через пробел записаны два целых числа  $M$  и  $N$  ( $1 \leq M, N \leq 32767$ ), размеры исследуемого прямоугольного участка.

### Формат выходных данных

Программа должна вывести одно целое число — количество поворотов, которое выполнит планетоход при исследовании заданной области на поверхности Марса.

### Примеры

<code>planet.in</code>	<code>planet.out</code>
3 4	5
5 3	6

## Задача С. Ходом коня

Имя входного файла: `knight.in`  
Имя выходного файла: `knight.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Как известно, в шахматах горизонтальные строки обозначаются цифрами от 1 до 8, считая от расположения белых фигур, стоящих внизу доски, а вертикальные столбцы — буквами латинского алфавита: А, В, С, D, E, F, G, H.

На шахматной доске в клетке с заданными координатами находится конь. Для клетки А1 после первого хода возможно перемещение коня на клетку С2 или В3.

Требуется написать программу, которая определяет координаты всех клеток, куда можно пойти конём первым ходом.

### Формат входных данных

В единственной входной строке записано обозначение исходной позиции коня на шахматной доске.

### Формат выходных данных

В единственной строке должны быть записаны через пробел обозначения всех клеток, в которые может переместиться конь после первого хода. Клетки выводятся в следующем порядке: вначале клетки первого ряда слева — направо, далее клетки второго ряда и т.д.

### Примеры

<code>knight.in</code>	<code>knight.out</code>
A1	C2 B3

## Задача D. Ходы коня-2

Имя входного файла: `knight2.in`  
Имя выходного файла: `knight2.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Как известно, в шахматах горизонтальные строки обозначаются цифрами от 1 до 8, считая от расположения белых фигур, стоящих внизу доски, а вертикальные столбцы — буквами латинского алфавита: A, B, C, D, E, F, G, H.

На шахматной доске в клетке с заданными координатами находится конь. Сначала делается первый ход конём, а затем — второй ход. Например, для клетки A1 после первого хода возможно перемещение коня на клетку C2 или B3, а после второго хода — на клетки A1, E1, A3, E3, B4, D4.

Требуется написать программу, которая определяет координаты всех клеток, куда можно прийти конём за два хода.

### Формат входных данных

В единственной входной строке записано обозначение исходной позиции коня на шахматной доске.

### Формат выходных данных

В единственной строке должны быть записаны через пробел обозначения всех клеток, в которые может переместиться конь после второго хода. Клетки выводятся в следующем порядке: вначале клетки первого ряда слева — направо, далее клетки второго ряда и т.д.

### Примеры

<code>knight2.in</code>	<code>knight2.out</code>
A1	A1 C1 E1 D2 A3 E3 B4 D4 A5 C5

## Задача Е. Газон фермера

Имя входного файла: `fermer.in`  
 Имя выходного файла: `fermer.out`  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 Мб

Английский фермер тщательно следит за своим газоном, в котором в каждой точке с целыми координатами растет один пучок травы. Как-то фермер воспользовался газонокосилкой и постриг траву на некотором прямоугольном участке газона. Стороны этого участка параллельны осям координат, а две противоположные вершины расположены в точках  $(x_1, y_1)$  и  $(x_2, y_2)$ . Следует отметить, что пучки травы, находящиеся на границе этого прямоугольника, также были пострижены.

Для полива газона фермер установил в точке с координатами  $(x_3, y_3)$  дождевальную установку, радиус действия которой равен  $r$ . Таким образом, установка начала поливать все пучки, расстояние от которых до точки  $(x_3, y_3)$  не превышало  $r$ .

Фермера заинтересовал следующий вопрос: сколько пучков травы оказалось и пострижено, и полито в этот день? Требуется написать программу, которая позволит дать ответ на вопрос фермера.



### Формат входных данных

Первая входная строка содержит четыре целых числа:  $x_1, y_1, x_2, y_2$  ( $-100000 \leq x_1 < x_2 \leq 100000$ ;  $-100000 \leq y_1 < y_2 \leq 100000$ ). Во второй входной строке записаны три целых числа:  $x_3, y_3, r$  ( $-100000 \leq x_3, y_3 \leq 100000$ ;  $1 \leq r \leq 100000$ ).

### Формат выходных данных

Необходимо вывести одно целое число — число пучков травы, которые были и пострижены, и политы.

### Система оценки и описание подзадач

#### Подзадача 1 (50 баллов)

Все координаты не превосходят 1000 по абсолютной величине.

#### Подзадача 2 (50 баллов)

Дополнительные ограничения отсутствуют.

### Примеры

<code>fermer.in</code>	<code>fermer.out</code>
0 0 5 4 4 0 3	14
-100 -100 100 100 0 0 50	7845
0 0 10 10 100 100 5	0

## Задача F. Троично-симметричная система

Имя входного файла: `system.in`  
Имя выходного файла: `system.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Числа в позиционной троично-симметричной системе счисления записываются с использованием трех символов: +, -, 0. Например, такими числами являются, например,

"++0-0" "--0+" "--".

Эти числа переводятся в десятичную систему как:

а)  $++0-0 = 1 * 3^4 + 1 * 3^3 + 0 * 3^2 - 1 * 3^1 + 0 * 3^0$

б)  $--0+ = -1 * 3^3 - 1 * 3^2 + 0 * 3^1 + 1 * 3^0$

в)  $--- = -1 * 3^2 - 1 * 3^1 - 1 * 3^0$

Над числами в позиционной троично-симметричной системе счисления можно выполнять два действия: сложение (+) и вычитание (-). Требуется написать программу, которая вычисляет сумму или разность чисел в троично-симметричной системе счисления. Таблица Пифагора для сложения цифр в троично-симметричной системе счисления имеет вид:

(+)	-	0	+
-	-+	-	0
0	-	0	+
+	0	+	+-

### Формат входных данных

В единственной строке записаны два числа в троично-симметричной системе счисления, между которыми в скобках записана требуемая операция. Разрядность чисел не превышает 15.

### Формат выходных данных

В единственной строке необходимо вывести полученный в результате заданной операции результат в троично-симметричной системе счисления без лидирующих нулей. Если ответ равен нулю выведите пустую строку.

### Примеры

<code>system.in</code>	<code>system.out</code>
<code>+++0-(+)-0+</code>	<code>++000</code>

## Задача G. Вирусы

Имя входного файла: `virus.in`  
Имя выходного файла: `virus.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Для моделирования различных объектов часто применяются так называемые клеточные поля. В простейшем случае — это прямоугольные таблицы, характеризующие некоторую область, а в каждой ячейке таблицы записывается какая-либо информация об исследуемом объекте. В биологии для моделирования распространения вирусов на плоской области в каждой ячейке помечается наличие вируса, а его распространение осуществляется в соседние ячейки по вертикали и горизонтали за одну единицу времени. Некоторые клетки обладают иммунитетом, заразить их невозможно и через них не распространяются вирусы.

Требуется написать программу, которая определяет минимально возможное число вирусов, с помощью которых можно заразить всю исследуемую прямоугольную область (за исключением защищённых клеток).

В приведённом примере таблица имеет размер  $4 \times 5$ , в ней символом "I" помечены защищённые клетки. Видно, что двух вирусов достаточно для заражения всей области. Их можно поместить, например, в клетки, помеченные символом "V".

V		I		
I	I			
			V	

### Формат входных данных

В первой входной строке записаны два натуральных числа  $M$  и  $N$  — размеры таблицы (количество строк и столбцов соответственно). Известно, что  $1 \leq M, N \leq 100$ . Во второй строке вначале записано одно число  $K$  — количество защищённых клеток, а далее записаны  $2K$  чисел — координаты этих клеток  $x_i, y_i$  ( $0 \leq k \leq M \cdot N$ ,  $1 \leq x_i \leq M$ ,  $1 \leq y_i \leq N$ ).

### Формат выходных данных

Программа должна вывести одно число — минимально возможное число вирусов.

### Примеры

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