

## Задача А. Четвертая точка

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Мише необходимо выбрать четыре точки на плоскости таким образом, чтобы получился прямоугольник со сторонами, параллельными осям координат. Он уже выбрал три точки и уверен, что не ошибся. Однако у него теперь возникла проблема нахождения последней четвертой точки. Поэтому Миша обратился к вам за помощью и просит написать программу, которая найдет недостающую точку прямоугольника.

### Формат входных данных

Данные задаются в трех строках. В каждой из них записано по два числа, определяющих абсциссу и ординату соответствующей точки. Все числа находятся в диапазоне от 1 до 1000.

### Формат выходных данных

Выведите два числа – координаты четвертой вершины прямоугольника.

### Примеры

<code>standard input</code>	<code>standard output</code>
4 4 6 4 4 6	6 6
18 13 5 5 5 13	18 5

## Задача В. Колышки

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Суть известной логической игры Колышки заключается в следующем. Есть поле в форме креста, на котором колышки расставлены следующим образом:

```
ooo
ooo
ooooooooo
ooo.ooo
ooooooooo
ooo
ooo
```

Символом 'o' обозначается колышек, а символом '.' - пустое поле. За один ход игрок может выбрать произвольным образом один колышек и одно из четырех направлений (вверх, вниз, влево или вправо), если в соответствующем направлении в соседней клетке находится еще один колышек и пустая клетка за ним. Тогда выбранный колышек перепрыгивает через соседний в соответствующем направлении и становится в клетку за ним, а колышек, через который был совершен прыжок, снимается с поля.

Цель игры – добиться того, чтобы на поле остался только один колышек.

В данной задаче **не требуется** находить последовательность действий игрока, которая поможет прийти к этой цели. Вам необходимо написать программу, определяющую для заданного расположения колышков на поле (позиции в игре) количество допустимых ходов, которые может сделать игрок из этой позиции.

### Формат входных данных

Задаются семь строк, каждая из которых содержит по семь символов (без учета переводов строк). Первые два и последние два символа первых двух и последних двух строк будут всегда пробелами. Оставшиеся символы - либо 'o' (строчная латинская буква), либо '.' (точка).

### Формат выходных данных

Выведите одно число – количество допустимых ходов в заданной позиции.

### Примеры

standard input	standard output
ooo ooo ooooooooo ooo.ooo ooooooooo ooo ooo	4
ooo ooo ..ooo.. oo...oo ..ooo.. ooo ooo	12

## Задача С. Пополнение в семействе

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Петя и Маша живут счастливо в браке. У них есть  $N$  сыновей, которым Петя и Маша дали имена в виде **различных четных** чисел  $b_1, b_2, \dots, b_N$ . Сейчас они ждут еще одного ребенка, который (они надеются на это) будет девочкой, и подбирают для нее подходящее красивое имя. Супруги договорились, что имя будет нечетным числом из отрезка  $[l, r]$ . Все нечетные числа в этом диапазоне кажутся им одинаково красивыми, но чтобы не путаться в именах Петя и Маша решили выбрать такое из чисел, которое максимизирует расстояние до ближайшего из  $N$  мальчиков.

Более точно, они выбирают нечетное число  $x \in [l, r]$  таким образом, чтобы выражение  $\min_{i=1, N} |x - b_i|$  было максимально возможным. Помогите Пете и Маше сделать оптимальный выбор.

### Формат входных данных

В первой строке задается целое число  $N$  ( $1 \leq N \leq 100$ ). Во второй строке –  $N$  различных четных чисел  $b_1, b_2, \dots, b_N$  ( $2 \leq b_i \leq 10^9$ ). Третья строка содержит два целых числа  $l$  и  $r$  ( $1 \leq l < r \leq 10^9$ ).

### Формат выходных данных

Выведите одно число – наиболее подходящее имя для девочки. Если есть несколько таких значений, можно вывести любое из них.

### Примеры

standard input	standard output
3 2 6 16 20 50	49
3 2 6 16 3 15	11
3 2 6 16 1 7	5

## Задача D. Скобочные последовательности

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Скобочной последовательностью называется строка символов, которая содержит только открывающие и закрывающие скобки трех типов:

- круглые: `()`
- квадратные: `[]`
- фигурные: `{}`

Скобочная последовательность считается правильной, если она удовлетворяет следующим условиям:

- Пустая строка является правильной скобочной последовательностью.
- Если  $A$  – правильная скобочная последовательность, то строки  $(A)$ ,  $[A]$ ,  $\{A\}$  также являются правильными скобочными последовательностями.
- Если  $A$  и  $B$  – правильные скобочные последовательности, то  $AB$  также является правильной скобочной последовательностью.

Например, скобочные последовательности `[({})]`, `[](){}`  и `{()}(){}`  являются правильными, а последовательности `[({([, []({)} и [{}])({)} – нет.`

Вася нашел строку, которая выглядит похожей на правильную скобочную последовательность. Однако некоторые символы в ней были записаны неразборчиво и могут оказаться какими угодно.

Помогите Васе подсчитать, сколько есть различных способов прочтения найденной строки как правильной скобочной последовательности. Более точно, сколькими способами можно заменить все неразборчивые символы в строке какими-либо скобками, так чтобы в результате получилась правильная скобочная последовательность. Так как это число может оказаться огромным, требуется найти только его последние (младшие) 5 десятичных цифр.

### Формат входных данных

Первая строка содержит четное число  $l$  ( $2 \leq l \leq 200$ ), определяющее длину найденной строки. Вторая строка содержит  $l$  символов (без пробелов), образующих найденную Васей строку. Неразборчивые символы представляются символом ‘?’.

### Формат выходных данных

Выведите последние пять цифр числа, определяющего количество способов прочтения строки.

### Примеры

standard input	standard output
6 <code>()()()</code>	1
10 <code>(?([?])?)?</code>	3
16 <code>???[????????]????</code>	92202

### Пояснение к примеру

Во втором примере будут подходящими следующие правильные скобочные последовательности: `{([()])}`, `()([()]{})` и `[[[]]]{}`.

## Задача Е. Белые шахматные фигуры

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Разбирая разные вещи на чердаке, Андрей нашел набор шахматных фигур. К сожалению, все эти фигуры оказались белыми и возможно их количество является неправильным. Правильный набор должен состоять из:

- одного короля;
- одного ферзя;
- двух ладей;
- двух слонов;
- двух коней;
- восьми пешек.

Андрей хотел бы узнать, сколько фигурок каждого типа он должен добавить или удалить из своего набора, чтобы получился правильный набор.

### Формат входных данных

В единственной строке задаются шесть целых чисел, в пределах от 0 до 10, которые определяют количество королей, ферзей, ладей, слонов, коней и пешек в найденном Андреем наборе.

### Формат выходных данных

Требуется вывести шесть чисел – количество фигур каждого типа, которые Андрей должен добавить или удалить. Положительные числа соответствуют добавляемым фигурам, а отрицательные – удаляемым.

### Примеры

<code>standard input</code>	<code>standard output</code>
0 1 2 2 2 7	1 0 0 0 0 1
2 1 2 1 2 1	-1 0 0 1 0 7

## Задача F. Шахматная доска

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

После того как Андрей нашел шахматные фигуры, он захотел поиграть в них. Но для этого Андрею необходима шахматная доска. К сожалению, шахматной доски у него не оказалось, но нашлась старая доска прямоугольной формы, которую Андрей решил самостоятельно разбить на клетки (прямоугольной формы, но необязательно квадратные).

Для этого он провел на доске  $N$  различных линий. Каждая линия может быть вертикальной или горизонтальной (параллельной одной из сторон доски) и должна проходить от одного края доски до противоположного.

Ваша задача – выяснить, какое максимальное количество клеток могло получиться при этом.

### Формат входных данных

В единственной строке задается одно целое число  $N$  ( $1 \leq N \leq 100$ ).

### Формат выходных данных

Выведите одно число – максимальное количество клеток, на которые может быть разбита доска  $N$  линиями.

### Примеры

<code>standard input</code>	<code>standard output</code>
1	2
3	6

## Задача G. Восстановление круга

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

По периметру круга были последовательно записаны некоторые  $N$  положительных целых чисел (за  $N$ -ым числом следует первое). Далее к каждому из чисел были добавлены значения двух соседних чисел (их первоначальные значения). То, что получилось в результате указанной операции, известно.

Требуется восстановить первоначальные значения в круге.

### Формат входных данных

В первой строке задается одно целое число  $N$  ( $3 \leq N \leq 10000$ ). Каждая из последующих  $N$  строк содержит целое число из диапазона от 1 до  $10^9$ . Эти числа в соответствующем порядке определяют получившиеся после добавления значения.

Гарантируется, что всегда существует решение, но не обязательно единственное.

### Формат выходных данных

Выведите в  $N$  строках первоначальные значения элементов в круге. Эти числа должны быть целыми положительными.

Не допускается вращение круга. Например, сумма первых трех выходных значений должна быть равна второму числу в круг, заданном во входных данных. Если существует несколько решений, можно вывести любое из них.

### Примеры

standard input	standard output
3	2
5	1
5	2
5	
4	5
20	8
15	2
17	7
14	
5	4
7	1
8	3
9	5
10	2
11	

## Задача Н. Переупорядочивание

Имя входного файла: `standard input`  
Имя выходного файла: `standard output`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Вам заданы три целых числа  $A$ ,  $B$  и  $C$ . Числа задаются не обязательно в этом порядке, но известно, что  $A$  меньше  $B$ , а  $B$  меньше  $C$ .

Чтобы представить их в более приятном виде, мы хотим их переупорядочить в заданном порядке.

### Формат входных данных

В первой строке задаются три положительных целых числа  $A$ ,  $B$ ,  $C$ , но не обязательно в таком порядке. Все три числа не превосходят 100. Вторая строка состоит из трех заглавных латинских букв 'A', 'B', 'C' и определяет требуемый порядок.

### Формат выходных данных

Выведите значения  $A$ ,  $B$  и  $C$  в требуемом порядке.

### Примеры

<code>standard input</code>	<code>standard output</code>
1 5 3 ABC	1 3 5
6 4 2 CAB	6 2 4