

## Задача 1. Боулинг

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Начинающий разработчик игр решил запрограммировать игру в кегли. Поскольку программировать трехмерную графику достаточно сложно, он решил сделать упрощенный вариант.

$N$  кеглей стоят на одной прямой, назовем её осью  $X$ . Для каждой кегли с номером  $i$  известна ее координата  $X_i$  и высота  $H_i$ . При броске мяч может задеть только одну кеглю, пусть это будет кегля с номером  $k$ . Если  $k$ -ая кегля падает вправо, она сбивает все другие кегли, находящиеся на отрезке  $[X_k, X_k + H_k]$ , при этом все сбитые кегли также падают вправо и сбивают все, которых коснутся и т.д. Если  $k$ -ая кегля падает влево, то все задетые кегли (на отрезке  $[X_k - H_k, X_k]$ ), аналогично, падают влево.

Для заданной расстановки кеглей определите, какое минимальное число бросков потребуется, чтобы уронить все кегли.

### Формат входных данных

В первой строке одно целое число  $N$  ( $0 < N \leq 10^5$ ). Во второй строке  $N$  пар целых чисел, все числа отделены друг от друга не менее чем одним пробелом. Первое число в каждой паре – координата кегли  $X_i$  ( $0 \leq X_i \leq 10^9$ ), второе – высота  $H_i$  ( $0 < H_i \leq 10^5$ ). Кегли перечислены в порядке возрастания координаты.

### Формат выходных данных

В единственной строке одно целое неотрицательное число  $X$  – искомое число бросков.

### Примеры

standard input	standard output
3 0 2 2 3 4 1	1

## Задача 2. Новогодние хлопоты

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Трехлетняя Кристина вернулась из детского сада в хорошем настроении и немедленно обрадовала папу, молодого программиста:

– Папа, скоро Новый Год!

Сергей испугано взглянул на календарь и, убедившись, что ноябрь еще не закончился, обреченно поинтересовался у жены:

– Что случилось?

– В садике было собрание, раздавали задания родителям к новомуднему утреннику.

– И что нам досталось?

– Всего-то сделать снежинки и подвесить их к потолку!

Сергей задумался. Белый пластик у него где-то был, остался от ремонта мебели. Трафареты снежинок в интернете можно найти, в крайнем случае жена нарисует. Но вот подвесить снежинку надо так, чтобы она висела горизонтально. А это уже проблема: не у любой снежинки возможная точка подвески попадет на пластик... Придётся писать программу определения точки подвески для каждого трафарета.

### Формат входных данных

Снежинка описывается внешним контуром и контурами отверстий. Каждый контур является замкнутой ломаной линией без самопересечений. Все контуры не имеют общих точек. Все отверстия лежат внутри внешнего контура, никакое отверстие не лежит внутри другого отверстия. Контуры определяются последовательностью вершин, которые соединяются отрезками прямых, в обходе по часовой стрелке. В первой строке файла указано число точек  $N_0$  во внешнем контуре снежинки,  $2 < N_0 \leq 40\,000$ . В следующей строке задано  $N_0$  пар целых чисел, все числа разделены пробелами, каждая пара чисел является декартовыми координатами  $x$  и  $y$  вершины контура, каждое число по модулю не превышает 10 000. В третьей строке задано количество отверстий  $K$ ,  $0 \leq K \leq 10\,000$ . Далее следует  $K$  пар строк, описывающих контуры отверстий, в первой строке каждой пары указывается число точек  $N_i$  в  $i$ -ом отверстии снежинки,  $2 < N_i \leq 10\,000$ , в следующей строке задано  $N_i$  пар целых чисел, все числа разделены пробелами, каждая пара чисел определяет координаты вершины контура. Общее число точек во всех контурах в описании снежинки не превышает 50 000.

### Формат выходных данных

В случае, если точка подвеса снежинки лежит вне фигуры, вывести «NO».

В случае, если точка подвеса снежинки лежит внутри или на границе фигуры, в первой строке вывести «YES», а во второй строке вывести координаты этой точки в виде двух несократимых дробей, разделенных пробелом. Несократимая дробь – это два взаимно простых целых числа (числитель и знаменатель), разделенных знаком '/' без пробелов, второе из которых является положительным. Для целого значения координаты знаменатель дроби следует выдать равным 1.

### Примеры

standard input	standard output
4 -1 -1 -1 3 2 3 2 -1 0	YES 1/2 1/1
4 -1 -1 -1 3 2 3 2 -1 1 4 0 0 0 2 1 2 1 0	NO

## Задача 3. Апельсин – спаниель

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Анаграммой некоторого слова является слово, полученное из исходного перестановкой его букв. Особый интерес представляют пары анаграмм, в которых каждое слово является осмысленным. Для нахождения всех таких пар разрабатывается программное обеспечение, которое по заданному словарю будет перебирать все возможные пары слов и проверять, являются ли они анаграммами друг друга. Вам предлагается написать основной модуль для этого проекта.

### Формат входных данных

В первой и во второй строках по одному слову. Каждое слово представляет собой непустую последовательность из строчных латинских букв, длина которой не превышает  $10^5$  символов.

### Формат выходных данных

В первой и единственной строке «YES», если исходные слова являются парой анаграмм, и «NO» в противном случае.

### Примеры

standard input	standard output
glfyk gylkf	YES
glfeyk grylkf	NO

## Задача 4. Без сдачи

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Все периодически сталкиваются с ситуацией, когда у кассира нет сдачи и приходится искать по карманам мелочь, чтобы набрать в точности нужную сумму. Чтобы избежать тех случаев, когда несмотря на все усилия, собрать требуемую сумму так и не удастся, можно всегда иметь при себе такой набор денежных знаков разного номинала, чтобы из него точно можно было набрать любую сумму (разумеется, до какого-то разумного предела). При этом, конечно, желательно, чтобы общее количество денежных знаков было минимально возможным.

Зная все возможные номиналы, Вам нужно определить, какое минимальное количество денежных знаков необходимо для того, чтобы набрать из них любую сумму от 1 до некоторого заданного числа включительно.

### Формат входных данных

В первой строке одно целое положительное число  $N$  – количество возможных номиналов денежных единиц,  $1 \leq N \leq 1000$ .

Во второй строке  $N$  различных целых положительных чисел через пробел – возможные номиналы. Каждое из чисел не превышает  $10^6$ . Гарантируется, что одно из этих чисел – 1.

В третьей строке одно целое положительное число  $S$ ,  $1 \leq S \leq 10^{10}$ .

### Формат выходных данных

В первой и единственной строке одно целое положительное число – минимальное количество денежных знаков доступных номиналов, достаточное для того, чтобы набрать любую сумму от 1 до  $S$  включительно.

### Примеры

standard input	standard output
4 1 3 4 7 7	4

## Задача 5. Запоминание номеров

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Многие сталкивались с необходимостью при заполнении документов указывать длинные номера (телефонов, паспорта, зачетки...), которые трудно запомнить. Однако для программистов можно придумать способ, с помощью которого номера будет запоминать немного интереснее. Надо всего лишь подобрать такую систему счисления, в которой этот номер будет иметь минимальное количество ненулевых разрядов. Естественно, хотелось бы получить минимальное из всех возможных оснований, чтобы его было легче запомнить.

### Формат входных данных

В единственной строке целое положительное число  $N$ ,  $2 \leq N \leq 10^{15}$ .

### Формат выходных данных

В единственной строке целое положительное число – минимальное основание системы счисления, в которой  $N$  имеет минимальное количество ненулевых разрядов.

### Примеры

standard input	standard output
6	3

## Задача 6. Стрелки

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Рассмотрим механические часы с двумя стрелками (часовой и минутной), которые движутся дискретно и поворачиваются по истечении каждой секунды на фиксированный угол, показывая в этот момент абсолютно точное время. Часовая стрелка при этом проходит полный оборот за 12 часов, а минутная – за 1 час.

Зная время с точностью до секунды, требуется определить угол между стрелками.

### Формат входных данных

В первой строке и единственной строке время в формате `hh:mm:ss` ( $00 \leq hh \leq 23$ ,  $00 \leq mm \leq 59$ ,  $00 \leq ss \leq 59$ ).

### Формат выходных данных

В первой и единственной строке три целых неотрицательных числа через пробел  $D$ ,  $M$ ,  $S$  – градусная мера угла между часовой и минутной стрелкой в заданный момент времени с точностью до секунды ( $D$  – градусы,  $M$  – минуты,  $S$  – секунды), принадлежащая диапазону от 0 до 180 градусов включительно. Напомним, что минута равна  $1/60$  градуса, а секунда –  $1/60$  минуты.

### Примеры

standard input	standard output
01:02:03	18 43 30

## Задача 7. Долой преграды

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 MiB

Один фермер решил выделить каждому из своих животных отдельный участок луга, где они будут пастись. Для этого он вбил в землю несколько столбиков и протянул между ними проволоку разной толщины. Таким образом он получил несколько участков, некоторые из которых граничили между собой, причем на каждом участке паслось ровно одно животное. Отметим, что каждый участок в объединении со своей границей представляет собой замкнутую односвязную область.

Животным такое нововведение не понравилось, ведь они привыкли пастись вместе, потому они решили разрушить ограждения так, чтобы иметь возможность собраться на любом из заданных участков. Определите, какова минимальная суммарная толщина проволоки, которую им для этого следует перегрызть.

### Формат входных данных

В первой строке одно целое положительное число  $N$  – количество областей,  $1 \leq N \leq 90\,000$ .

Далее  $N$  блоков по 3 строки в каждом – описание областей.

В первой строке  $i$ -го блока одно целое положительное число  $K_i$  – количество столбиков (и, соответственно, сегментов проволоки) на границе  $i$ -ой области,  $3 \leq K_i \leq 600$ .

Во второй строке  $i$ -го блока  $K_i$  различных целых неотрицательных чисел через пробел – номера столбиков, входящих в границу  $i$ -ой области в порядке обхода по часовой стрелке

В третьей строке  $i$ -го блока  $K_i$  целых неотрицательных чисел через пробел  $W_{i,j}$  – толщина проволоки между соответствующими столбиками границы,  $1 \leq W_{i,j} \leq 100\,000$ .  $W_{i,1}$  соответствует участку между первым и вторым столбиком,  $W_{i,2}$  – между вторым и третьим, и так далее.  $W_{i,K_i}$  соответствует участку границы между последним и первым столбиками.

Гарантируется, что номера столбиков не превышают 5 000 000, а их общее количество не превышает 600 000. Также гарантируется, что если один участок границы принадлежит двум областям, то указанная для него толщина проволоки будет одинаковой. Гарантируется, что каждый участок границы принадлежит не более чем двум подобластям.

### Формат выходных данных

В первой и единственной строке одно целое неотрицательное число – суммарная толщина проволоки, которую необходимо перегрызть, чтобы каждое животное имело возможность пройти на любой из исходных участков.

### Примеры

standard input	standard output
3	15
4	
1 2 3 4	
4 4 1 4	
4	
4 3 9 10	
1 4 4 4	
3	
14 15 16	
10 10 10	

## Задача 8. В центр

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

В центре города N много лет стоял красивый металлический куб с длиной ребра 1 метр, ориентированный по сторонам света. На стороне, обращенной к северу, была выгравирована буква Р, к югу – Г, к западу – Д и к востоку – К. Накануне очередного юбилея была составлена уточненная карта этого города, и выяснилось, что центр города немного смещен. Совет города принял решение экстренно переместить куб в «настоящий» центр. Попытки просто двигать его по земле успехом не увенчались – и куб слишком тяжелый, и асфальт не скользкий. Крановщик, которого позвали, чтобы поднять куб и переставить его на нужное место, сказал, что не в форме и до утра следующего дня он вряд ли справится со столь ответственной задачей. В сложившейся ситуации остался единственный возможный выход – перекачивать куб через ребра, которыми он касается земли. К счастью, сместить куб надо было на целое число метров в обоих направлениях, а потому такой метод не казался бесперспективным.

Помогите горожанам определить, действительно ли им удастся перекатить куб в требуемое место и, если да, какое минимальное количество переворотов им необходимо будет совершить. Разумеется, стороны куба в результате должны сохранить свою ориентацию.

### Формат входных данных

В первой строке два целых числа через пробел – на сколько метров необходимо переместить куб в направлении север-юг и восток-запад. Смещениям на север и восток соответствуют положительные числа, а на юг и запад – отрицательные.

Каждое из чисел по модулю не превышает  $10^9$ .

### Формат выходных данных

В первой и единственной строке одно целое число – минимальное количество переворотов куба, за которое его можно переместить в заданную точку с сохранением ориентации. Если заданное перемещение невозможно, вывести  $-1$ .

### Примеры

standard input	standard output
2 4	6
2 5	-1



## Задача 9. Можно и так

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Мы привыкли в качестве игральных кубиков использовать кубики, граням которых присвоены значения от 1 до 6. Используя два таких кубика и полагая результатом броска сумму значений на верхних гранях, можно выбросить произвольное значение от 2 до 12 с соответствующей ненулевой вероятностью.

Обобщим понятие игрального кубика следующим образом. Будем называть игральным кубиком кубик, граням которого приписаны произвольные целые неотрицательные значения. Два набора игральных кубиков назовем эквивалентными, если они состоят из равного количества кубиков и вероятности бросков с любым произвольным результатом для них совпадают (результатом броска здесь по-прежнему считается сумма значений на верхних гранях кубиков). Удивителен тот факт, что даже для «стандартного» набора из двух игральных кубиков существует несколько наборов, эквивалентных ему.

Вам необходимо для заданного набора из двух игральных кубиков определить количество всех различных наборов, эквивалентных ему. Два набора различны, если между их кубиками нельзя установить взаимно-однозначное соответствие, при котором все соответствующие кубики совпадают. Два кубика совпадают, если один можно получить из другого перестановкой значений на его гранях.

### Формат входных данных

В первой строке 6 целых неотрицательных чисел через пробел – значения на гранях первого кубика.

Во второй строке 6 целых неотрицательных чисел через пробел – значения на гранях второго кубика.

Все числа не превышают 1000.

### Формат выходных данных

В первой и единственной строке одно целое положительное число – количество различных наборов игральных кубиков, эквивалентных заданному (включая сам исходный набор).

### Примеры

standard input	standard output
1 2 3 4 5 6	5
1 2 3 4 5 6	

## Задача 10. Код

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Рассмотрим кодовый замок, внешняя часть которого состоит из  $N$  цилиндров, на каждом из которых можно выставить цифры от 0 до 9. Для каждого кода, представляющего собой последовательность цифр длины  $N$ , введем понятие контрольного значения, которое будет вычисляться по следующей схеме:

- 1) Пусть  $A$  – исходный код длины  $N$ .
- 2) Если длина  $A$  равна 1, то  $A$  – контрольное значение.
- 3) Пусть  $L$  – длина кода  $A$ . Сформировать код  $B$  длины  $(L - 1)$ ,  
 $B_i := (A_i + A_{i+1}) \bmod 10, i = 1, 2, \dots, (L - 1)$ .
- 4)  $A := B$ .

Таким образом, контрольное значение является некоторой цифрой от 0 до 9 включительно.

Также введем понятие номера кода, определяемое следующим образом. Для кода вычисляется его контрольное значение, затем все возможные коды той же длины и с тем же контрольным значением упорядочиваются в лексикографическом порядке и нумеруются, начиная с 1. Тот номер, который в результате получает исходный код, и есть искомый.

Вам требуется, зная контрольное значение кода и номер кода, определить сам код.

### Формат входных данных

В первой строке две целых числа через пробел:  $X$  – контрольное значение кода,  $0 \leq X \leq 9$ ,  $N$  – длина кода,  $1 \leq N \leq 10^6$ .

Во второй строке одно целое положительное число  $K$  – номер кода,  $1 \leq K \leq 10^{1000000}$ .

### Формат выходных данных

В первой и единственной строке последовательность цифр длины  $N$  – код, имеющий заданное контрольное значение  $X$  и заданный номер  $K$ . Если такой код не существует, вывести  $-1$ .

### Примеры

standard input	standard output
5 2 3	23
5 2 15	-1

## Задача 11. Не нарушай

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

На особо опасном прямом участке дороги длины  $L$  решили установить систему контроля движения, включающую в себя  $N$  датчиков. Места для установки датчиков были выбраны произвольно по всей длине участка. После того, как места для датчиков были оборудованы, оказалось, что датчики должны постоянно обмениваться информацией. Согласно инструкции, для корректной работы системы необходимо, чтобы расстояние между любыми двумя датчиками не превышало  $D$ .

Определите вероятность того, что произвольно выбранные места под датчики удовлетворяют этому условию.

### Формат входных данных

В первой строке и единственной строке три целых положительных числа через пробел:  $L$  – длина участка дороги,  $D$  – максимально допустимое расстояние между датчиками,  $N$  – количество датчиков,  $1 \leq L, D, N \leq 15$ ,  $D \leq L$ .

### Формат выходных данных

В первой и единственной строке два целых положительных числа через '/' – числитель и знаменатель несократимой дроби, равной вероятности того, что  $N$  произвольно выбранных мест для датчиков на участке дороги длины  $L$  расположены так, что расстояние между любыми двумя из них не превышает  $D$ .

### Примеры

standard input	standard output
3 1 2	5/9

## Задача 12. Трезубец

Имя входного файла: *standard input*  
Имя выходного файла: *standard output*  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 MiB

Простейшее изображение трезубца состоит из трех вертикальных отрезков-зубьев, нижний край которых лежит на горизонтальном отрезке-основании, из середины которого вниз идет вертикальный отрезок-рукоятка. Зубья на основании располагаются симметрично – слева, справа и посередине. Символом '#' будем обозначать элементы трезубца, а символом '.' – свободное место. Ниже приведен пример подобного изображения:

```
#....  
#...#  
#.#.#  
#####  
..#..  
..#..
```

Зафиксируем длину основания равной 5, а расстояния между зубьями и толщину всех элементов равными 1. Изменяя оставшиеся параметры (длины зубьев и длину рукоятки), можно получить различные изображения трезубцев.

Необходимо написать программу, которая по заданным значениям параметров сформирует из символов '#' и '.' соответствующее изображение трезубца.

### Формат входных данных

В первой строке и единственной строке четыре целых положительных числа через пробел – длины левого, центрального и правого зубьев и длина рукоятки. Каждое из значений не превышает 200.

### Формат выходных данных

Минимальное возможное количество строк по 5 символов ('#' или '.'), составляющих изображение трезубца с заданными параметрами по указанным выше правилам.

### Примеры

standard input	standard output
3 1 2 2	#.... #...# #.#.# ##### ..#.. ..#..