

## Задача А. Середина игры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

А Вы знали, что сегодня проходит финал международного турнира по крестикам-ноликам на бесконечной доске? Вадим тоже не знал, но, услышав об этом монументальном событии года, он открыл стрим и увидел, что между собой сражаются два игрока, Алиса и Боб. Сейчас у Алисы  $A$  очков, а у Боба —  $B$  очков. Как всем интересующимся известно, за победу в раунде человеку даётся 2 очка, за ничью — 1 очко, за поражение — 0 очков.

Вадим очень опечален тем, что пропустил часть этой игры, ведь ему очень интересно, в скольких раундах победила Алиса, в скольких победил Боб и в скольких была ничья. Помогите ему ответить на этот вопрос.

### Формат входных данных

В первой строке дано одно целое число  $A$  — количество очков у Алисы ( $0 \leq A \leq 10^9$ ).

Во второй строке дано одно целое число  $B$  — количество очков у Боба ( $0 \leq B \leq 10^9$ ).

### Формат выходных данных

Если в данных есть ошибка, то есть текущие результаты никак не достижимы при проведении любого количество раундов, выведите «**Error**».

Если однозначно ответить невозможно, выведите «**Undefined**».

Иначе, выведите три числа — количество раундов, в которых победила Алиса, количество раундов, в которых победил Боб, и количество раундов, в которых была ничья.

### Примеры

стандартный ввод	стандартный вывод
3 1	1 0 1
2 2	Undefined
1 2	Error

## Задача В. Поп-ит

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

*Поп-ит* (с англ. *pop it* — «лопни это») — кнопочная игрушка, получившая популярность среди детей и подростков весной 2021 года. Представляет собой резиновую или силиконовую игрушку с полусферами для нажатия, при котором издаётся щёлкающий звук. Эти полусферы получили название «пупырок» по аналогии с пузырьчатой упаковкой, в которой похожие полусферы можно лопать со звуком. (Википедия)

Поп-ит в нашей задаче представляет собой прямоугольник высоты  $h$  и ширины  $w$ , каждая единичная клетка которого содержит пупырку. Каждая пупырка может быть в одном из двух состояний — выдавлена наружу или внутрь. Будем помечать состояние «выдавлена наружу» цифрой 0, а «выдавлена внутрь» — цифрой 1. Любую пупырку можно переключить, и тогда она поменяет своё состояние на противоположное.

Даша хочет привести поп-ит к исходному виду, в котором он продавался — а для этого нужно, чтобы все пупырки были в одинаковом состоянии. Какое наименьшее число пупырок Даше нужно для этого переключить?

### Формат входных данных

В первой строке даны два целых числа  $h$  и  $w$  — высота и ширина поп-ита ( $1 \leq h, w \leq 100$ ).

Каждая из следующих  $h$  строк содержит  $w$  символов 0 или 1 без пробелов. Эти строки описывают состояние пупырок на поп-ите.

### Формат выходных данных

Выведите одно целое число — наименьшее число пупырок, которое надо переключить, чтобы все пупырки оказались в одинаковом состоянии.

### Примеры

стандартный ввод	стандартный вывод
2 3 010 001	2
5 4 1111 1010 1101 1010 0111	6

## Задача С. Ужин для интровертов

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

У Марго очень много друзей и все они — интроверты. У него дома есть круглый стол, вокруг которого стоит  $N$  стульев. Как известно, интроверт за столом чувствует себя комфортно, если количество свободных стульев до ближайшего по кругу соседа слева и до ближайшего по кругу соседа справа от него в сумме хотя бы  $K$ . Марго хочет пригласить своих друзей на ужин. Для этого ему нужно узнать, какое наибольшее число интровертов можно посадить за стол так, чтобы каждому было комфортно. Помогите ему.

### Формат входных данных

В первой строке вводится целое число  $N$  — количество мест за столом ( $3 \leq N \leq 10^9$ ).

В следующей строке вводится целое число  $K$  — количество свободных стульев, необходимое для комфорта интроверта ( $1 \leq K \leq N - 2$ ).

### Формат выходных данных

Выведите единственное число — наибольшее число интровертов, которых можно посадить за стол. Гарантируется, что хотя бы двух человек можно посадить за стол.

### Примеры

стандартный ввод	стандартный вывод
5 2	2
6 1	4

## Задача D. Плохие ставки

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Два игрока играют в казино в следующую игру. Рулетка крутится  $N$  раз. Каждый раз на ней равновероятно выпадает число от 1 до  $K$ . После чего все числа суммируются. Тот, кто угадает итоговую сумму, победил.

Первый игрок поставил на то, что сумма в конце будет равна  $S_1$ , а второй на то, что сумма будет равна  $S_2$ . Определите у кого больше шансов победить.

### Формат входных данных

В первой строке дано целое число  $N$  — число запусков рулетки ( $1 \leq N \leq 10^9$ ).

Во второй строке дано целое число  $K$  — количество чисел на рулетке ( $1 \leq K \leq 10^9$ ).

В третьей строке дано целое число  $S_1$  — сумма, на которую поставил первый игрок ( $1 \leq S_1 \leq 10^9$ ).

В четвертой строке дано целое число  $S_2$  — сумма, на которую поставил второй игрок ( $1 \leq S_2 \leq 10^9$ ).

### Формат выходных данных

Выведите «First», если шансов победить больше у первого игрока, «Second», если шансов победить больше у второго игрока, и «Equal», если шансы на победу у игроков равны.

### Пример

стандартный ввод	стандартный вывод
3	First
3	
7	
4	

## Задача E. Ключало

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Ключало состоит из  $N$  деталей, у каждой есть свой стандарт —  $i$ -я деталь должна весить  $s_i$  грамм. Если есть ключало, в котором  $i$ -я деталь весит  $a_i$  грамм, то можно посчитать её *отклонение* по формуле  $\frac{|a_i - s_i|}{s_i}$ . У всей же конструкции *отклонение* считается по формуле  $\sum \frac{|a_i - s_i|}{s_i}$ , то есть сумма *отклонений* каждой детали. Допустимое отклонение ключала по стандарту равно  $K$ .

Вам дано ключало. За одну минуту можно либо увеличить вес одной детали на 1 грамм, либо уменьшить вес одной детали на 1 грамм. За какое наименьшее время можно привести данное ключало к стандарту с отклонением не больше  $K$ ?

### Формат входных данных

В первой строке даны два целых числа  $N$  и  $K$  — количество деталей в ключало и допустимое отклонение ( $1 \leq N \leq 10^5, 0 \leq K \leq 10^9$ ).

Во второй строке даны  $N$  целых чисел  $s_i$  — вес деталей в стандарте ( $1 \leq s_i \leq 10$ ).

В третьей строке даны  $N$  целых чисел  $a_i$  — вес деталей в данном ключало ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите наименьшее количество минут, за которое можно привести данное ключало к стандарту с отклонением не больше  $K$ .

### Пример

стандартный ввод	стандартный вывод
3 1 1 2 1 2 4 3	3

### Замечание

В примере можно уменьшить вес первой и третьей детали до стандарта.

## Задача F. Маска для монстров

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Монстрам надо спать, но не у всех это легко получается. Так монстру Вадиму, который выглядит как выпуклый многоугольник из  $N$  вершин на плоскости, часто мешает свет. У Вадима есть  $N$  глаз, по одному в каждой вершине, и чтобы спокойно уснуть, ему понадобится маска для монстров, закрывающая все глаза. Маска для монстров — это произвольная линия, которая должна вплотную прилегать к каждому глазу и не проходить внутри монстра. В магазине есть самые разные маски, но Вадиму хватит наименьшей по длине. Какой длины будет эта маска?

### Формат входных данных

В первой строке дано единственное целое число  $N$  — количество глаз монстра ( $3 \leq N \leq 10^5$ ).

В каждой из следующих  $N$  строк через пробел даны по два целых числа  $x_i, y_i$  — координаты вершин многоугольника в порядке обхода против часовой стрелки ( $-10^9 \leq x_i, y_i \leq 10^9$ ).

Гарантируется, что многоугольник выпуклый.

### Формат выходных данных

Выведите наименьшую длину маски, подходящей Вадиму.

Ответ будет засчитан, если его абсолютная или относительная погрешность не превосходит  $10^{-6}$ .

### Пример

стандартный ввод	стандартный вывод
4 0 0 2 0 2 2 0 2	6.000000

## Задача G. Наибольший наибольший общий делитель

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

*Все встречали задачи с ленивым условием. Эта задача не является исключением.*

Найдите пару различных целых чисел с наибольшим наибольшим общим делителем среди всех пар различных  $x, y$  таких, что  $L \leq x, y \leq R$ .

### Формат входных данных

Даны два целых числа  $L$  и  $R$  ( $1 \leq L < R \leq 10^{18}$ ,  $R - L \leq 10^6$ ).

### Формат выходных данных

Выведите пару различных целых чисел с наибольшим наибольшим общим делителем. Если таких пар несколько, выведите любую из них.

### Пример

стандартный ввод	стандартный вывод
4 13	6 12

## Задача Н. Прогрессивный NoSQL

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

«Гигачат» — прогрессивная IT-компания, разрабатывающая одноимённую прогрессивную социальную сеть. Недавно Ваня устроился работать в эту компанию. Его текущая задача — реализовать базу данных, которая будет отслеживать регистрации новых пользователей. К сожалению, сейчас Ваня готовится к четвертьфиналу ICPC, поэтому ему нужна Ваша помощь.

Изначально база данных пуста. В течение некоторого времени  $Q$  человек делают по одному запросу в эту базу данных. Каждый запрос представляет собой непустую строку, которая может содержать строчные и заглавные буквы английского алфавита, а также цифры. Длина строки не превышает 10. Если в системе ещё нет пользователя, именем которого является данная строка, то необходимо зарегистрировать пользователя с данным именем. В противном случае, к имени пользователя дописывается 1 и проверяется наличие такого пользователя в системе. Если такого пользователя нет, то необходимо зарегистрировать пользователя с данным именем (с единицей в конце). Если же такой пользователь уже есть, необходимо вместо 1 дописать 2, затем (в случае наличия такого пользователя) вместо 2 дописать 3 и так далее. Ответом на каждый запрос является полученное пользователем имя.

### Формат входных данных

В первой строке дано число  $Q$  ( $1 \leq Q \leq 10^5$ ) — количество запросов. Запросы даны в хронологическом порядке.

В каждой из последующих  $Q$  строк дана непустая строка, которая может содержать только строчные и заглавные буквы английского алфавита, а также цифры. Длина строки не превосходит 10.

### Формат выходных данных

Выведите  $Q$  строк, где  $i$ -я строка — имя, полученное  $i$ -м пользователем.

### Пример

стандартный ввод	стандартный вывод
11	fst
fst	lem
lem	fst1
fst	fst2
fst	margot
margot	Margot
Margot	margot1
margot	fst21
fst2	lem1
lem1	lem11
lem1	lem2
lem	

## Задача I. Марго покидает Мегабайтбург

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В связи с некоторыми событиями Марго необходимо покинуть Мегабайтбург. Известно, что этот город представляет собой прямоугольную матрицу, длина которой равна  $M$ , а высота –  $N$ . Клетки матрицы бывают двух типов: свободные (обозначаются символом ‘.’) и занятые стеной (обозначаются символом ‘#’). Марго может за один ход переместиться из клетки  $(i, j)$  в любую из клеток  $(i - 1, j)$ ,  $(i + 1, j)$ ,  $(i, j - 1)$ ,  $(i, j + 1)$ . Также Марго может не более  $K$  раз совершить в качестве хода *Мегапрыжок*: из клетки  $(i, j)$  попасть в любую из клеток  $(i - 2, j)$ ,  $(i + 2, j)$ ,  $(i, j - 2)$ ,  $(i, j + 2)$ . При этом, вне зависимости от того, использовался ли *Мегапрыжок* или нет, Марго должен завершить свой ход в свободной клетке, которая находится внутри Мегабайтбурга. Общежитие, в котором сейчас находится Марго, расположено в клетке  $(d_x, d_y)$ , а аэропорт, в который Марго хочет попасть, – в клетке  $(a_x, a_y)$ . Гарантируется, что общежитие и аэропорт находятся в разных свободных клетках. Сейчас нет времени на размышления, поэтому требуется Ваша помощь. Выясните, может ли Марго добраться от общежития до аэропорта.

### Формат входных данных

В первой строке даны числа  $N, M, K$  ( $2 \leq N, M \leq 1000, 0 \leq K \leq 10^6$ ) – размеры Мегабайтбурга и количество доступных Марго *Мегапрыжков*.

В каждой из последующих  $N$  строк дано  $M$  символов ‘.’ или ‘#’ – описание Мегабайтбурга.

В  $N + 2$ -й строке даны числа  $d_x, d_y$  ( $1 \leq d_x \leq N, 1 \leq d_y \leq M$ ) – координаты общежития. Гарантируется, что данная клетка свободна.

В последней строке даны числа  $a_x, a_y$  ( $1 \leq a_x \leq N, 1 \leq a_y \leq M$ ) – координаты аэропорта. Гарантируется, что данная клетка свободна.

Гарантируется, что координаты общежития не совпадают с координатами аэропорта.

### Формат выходных данных

Выведите «YES», если Марго может попасть из общежития в аэропорт. В противном случае выведите «NO». Ответ можно выводить в любом регистре.

### Примеры

стандартный ввод	стандартный вывод
3 3 1 .#. #.# .#. 1 1 3 3	NO
2 2 0 #. .. 1 2 2 1	YES
2 5 2 .#.#. ###.. 1 1 2 4	YES

### Замечание

Решение на языке Python можно ускорить, если отправить его на PyPy.

## Задача J. Snakes&Snakes

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Вадима есть одномерная доска для игры в Snakes&Snakes, состоящая из  $N$  клеток, которые пронумерованы от 1 до  $N$  слева направо. Изначально в клетке 1 стоит фишка. Цель игры — попасть в клетку  $N$ . Каждой клетке (кроме клеток 1 и  $N$ ) соответствует некоторое целое неотрицательное число  $p_i$ . Если  $p_i = 0$ , то  $i$ -я клетка пустая. В противном случае в клетке стоит телепорт, отправляющий фишку влево. Гарантируется, что клетки 1 и  $N$  пустые.

В Snakes&Snakes ход совершается по следующему алгоритму.

1. Игрок бросает шестигранный кубик. Если ему выпало число  $k$ , то он двигает фишку на  $k$  клеток вправо, при этом фишка не может оказаться правее клетки  $N$ . Другими словами, если фишка стояла в клетке  $i$ , то она оказывается в клетке  $\min(i + k, N)$ ;
2. Если фишка оказалась в клетке  $N$ , то игрок побеждает;
3. Если фишка оказалась в  $i$ -й клетке, которая не содержит телепорт ( $p_i = 0$ ), то происходит переход к шагу 4. В противном случае фишка перемещается влево на  $p_i$  клеток (в клетку с номером  $i - p_i$ ), после чего повторяется шаг 3;
4. Если игрок на шаге 1 выбросил на кубике 6, то он может повторить все действия алгоритма, начиная с шага 1, не прекращая текущий ход. В противном случае текущий ход игрока завершается.

Марго интересуется у Вадима, за какое минимальное количество ходов можно победить в этой игре (даже если это маловероятно). Помогите Вадиму ответить на данный вопрос.

### Формат входных данных

В первой строке дано число  $N$  ( $2 \leq N \leq 2 \cdot 10^5$ ) — размер доски.

Во второй строке даны  $N - 2$  числа  $p_i$  ( $0 \leq p_i < i, 1 < i < N$ ) — описание доски.

### Формат выходных данных

Выведите одно число — минимальное число ходов, необходимое для победы. Если добраться до клетки  $N$  нельзя, то выведите  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
10 0 1 1 1 1 1 1 0	-1
10 1 2 1 2 0 1 1 1	1
10 1 1 2 2 0 6 7 8	2

## Задача К. Бинарные деревья

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Вчера Вадим нашёл на дороге бинарное дерево  $a$  с корнем в 0 из  $N$  вершин. Однако любимым у него является бинарное дерево  $b$  с корнем в 0 из  $N$  вершин. Поэтому он решил преобразовать дерево  $a$  в дерево  $b$ , используя следующую операцию:

- Выбирается произвольная вершина  $v$ , кроме корня. Её поддерево, включая саму вершину, переподвешивается за другую вершину  $u$ , которая не принадлежит выбранному поддереву. Результатом должно получиться бинарное дерево с корнем в 0.

Вадим уверен, что с помощью подобной операции возможно привести найденное бинарное дерево в изоморфное его любимому, используя не более, чем  $N$  преобразований. Помогите ему найти последовательность этих преобразований.

Напомним, что *бинарное дерево* — это такое дерево, что каждая вершина является предком не более, чем 2 других вершин, у корня предка нет. Два корневых бинарных дерева называются *изоморфными*, если:

- Эти два дерева состоят из одной вершины;
- Количество детей у корней этих деревьев одинаковое, поддерево каждого ребёнка первого изоморфно поддереву какого-то ребёнка второго и поддерево каждого ребёнка второго изоморфно поддереву какого-то ребёнка первого.

### Формат входных данных

В первой строке дано целое число  $N$  — количество вершин в найденном и любимом деревьях ( $2 \leq N \leq 10^3$ ).

Во второй строке даны  $N - 1$  целых чисел  $pa_i$  — предки вершин найденного дерева с номерами от 1 до  $N - 1$  ( $0 \leq pa_i \leq N - 1$ ).

Во третьей строке даны  $N - 1$  целых чисел  $pb_i$  — предки вершин любимого дерева с номерами от 1 до  $N - 1$  ( $0 \leq pb_i \leq N - 1$ ).

Гарантируется, что данные деревья бинарные.

### Формат выходных данных

В первой строке выведите целое число  $M$  — количество использованных операций ( $0 \leq M \leq N$ ).

В следующих  $M$  строках выведите пары чисел  $v$  и  $u$  — корень выбранного поддерева и вершина, за которую это поддерево подвешивается во время текущей операции ( $1 \leq v \leq N - 1, 0 \leq u \leq N - 1$ ). Вершина  $u$  не может находиться в поддереве вершины  $v$ . Полученное после каждой операции дерево должно быть бинарным.

Гарантируется, что ответ существует. Если ответов несколько, выведите любой из них.

### Примеры

стандартный ввод	стандартный вывод
4 0 0 1 0 1 2	1 2 3
4 2 0 0 0 3 0	0

## Задача L. Апокалипсис

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

На планете Мидав очень близок конец света. Как известно, эта плоская планета, которую можно представить как бесконечную плоскость с декартовыми координатами. На этой планете есть  $Q$  поселений.

В нулевой день на Мидаве случилось заражение. Оно представляет из себя выпуклый многоугольник на  $N$  вершинах. Каждый день площадь заражения меняется неизвестным образом, но для каждого дня с номером  $i > 0$  верно следующее:

1. Если в  $i$ -й день заражена любая точка на расстоянии  $d$  от исходного многоугольника, то заражены и все остальные точки на расстоянии не большем  $d$  от исходного многоугольника;
2. Пусть  $S_k$  — площадь заражения в  $k$ -й день. Тогда верно  $S_i = 2 \cdot S_{i-1}$ .

Если какое-то поселение окажется внутри или на границе заражения, то все живые организмы в нём сразу же вымрут. Для каждого поселения планеты Мидав осталось совсем немного времени, поэтому ответьте, какой день (включая и нулевой) окажется для поселения последним.

### Формат входных данных

В первой строке дано целое число  $N$  — количество точек в многоугольнике заражения нулевого дня ( $3 \leq N \leq 10^5$ ).

В следующих  $N$  строках даны по два целых числа  $c_{xi}$  и  $c_{yi}$  — координаты вершин заражения.

В следующей строке дано целое число  $Q$  — количество поселений на Мидаве ( $1 \leq Q \leq 10^5$ ).

В следующих  $Q$  строках даны по два целых числа  $t_{xi}$  и  $t_{yi}$  — координаты каждого из поселений.

Все координаты по модулю не превосходят  $10^9$ . Гарантируется, что данный многоугольник выпуклый, а также, что вершины заданы в порядке обхода против часовой стрелки. Гарантируется, что поселения находятся на расстоянии не меньшем  $10^{-6}$  от границы заражения в любой из дней, кроме нулевого.

### Формат выходных данных

Выведите  $Q$  целых чисел — последние дни для поселений в порядке ввода.

### Пример

стандартный ввод	стандартный вывод
4	0
1 3	0
1 1	2
3 1	4
3 3	
4	
2 2	
1 2	
4 1	
6 2	

### Замечание

В примере второе поселение будет заражено в нулевой день, так как лежит на границе заражения.