

Задача А. Восстановление чисел

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

У Павла были два целых положительных числа a и b . Он нашел их сумму s и наибольший общий делитель g , после чего забыл a и b . Помогите ему восстановить исходные числа.

Формат входных данных

В единственной строке даны два целых числа s и g ($1 \leq s \leq 10^9, 1 \leq g \leq 10^9$) — сумма и наибольший общий делитель чисел a и b .

Формат выходных данных

Если Павел ошибся, и таких чисел a и b не существует, выведите одно число -1 .

Иначе выведите два целых положительных числа a и b в одной строке через пробел. Если существует несколько возможных решений, разрешается вывести любое.

Примеры

стандартный ввод	стандартный вывод
6 2	4 2
7 2	-1

Задача В. Минимальная площадь

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

Дан строго выпуклый многоугольник. Найдите минимально возможную площадь невырожденного треугольника, вершины которого являются вершинами многоугольника.

Формат входных данных

В первой строке дано целое число n ($3 \leq n \leq 200000$) — количество вершин многоугольника.

В каждой из следующих n строк даны два целых числа x_i и y_i ($-10^9 \leq x_i, y_i \leq 10^9$) — координаты вершин многоугольника.

Гарантируется, что многоугольник строго выпуклый. Вершины даны в порядке обхода против часовой стрелки.

Формат выходных данных

Известно, что площадь треугольника, вершины которого являются целыми точками на координатной плоскости — либо целое, либо полуцелое число.

Выведите целое число — искомую площадь, умноженную на 2.

Примеры

стандартный ввод	стандартный вывод
4 0 1 3 0 3 3 -1 3	5
3 0 0 1 0 0 1	1
4 -999999991 999999992 -999999993 -999999994 999999995 -999999996 999999997 999999998	39999999948000000156

Замечание

Рекомендуется все вычисления проводить в целых числах, так как точности чисел с плавающей точкой, скорее всего, не хватит.

Задача C. Third-Party Software

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

Павел разрабатывает игру. Для этого ему очень нужны функции, доступные в сторонней библиотеке, слишком известной, чтобы ее называть. Известно, что функция i впервые появилась в версии a_i и просуществовала вплоть до версии b_i , а в версии $b_i + 1$ ее уже не было в этой библиотеке.

Библиотека не бесплатная, а Павлу нужны все функции. Какое минимальное количество версий надо приобрести, чтобы можно было пользоваться всеми функциями?

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 200000$) — количество функций.

В каждой из следующих n строк дано два числа a_i и b_i ($1 \leq a_i \leq b_i \leq 10^9$) — промежуток версий библиотеки, в которых была доступна функция i .

Формат выходных данных

В первой строке выведите целое число k — минимальное количество версий библиотеки, которые нужно приобрести, чтобы можно было пользоваться всеми функциями.

Во второй строке выведите k различных целых чисел — номера версий, которые нужно приобрести.

Если существует несколько возможных ответов, разрешается вывести любой.

Пример

стандартный ввод	стандартный вывод
5	2
2 4	3 4
1 3	
2 3	
3 6	
4 5	

Задача D. Трансферное окно

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

Вы играете в футбольный менеджер. Всего в игре есть n футболистов, из них k футболистов — $a_1, a_2, \dots, a_k$ — в данный момент играют за вашу команду. Вы хотите, чтобы в вашей команде играли футболисты $b_1, b_2, \dots, b_k$. Для этого вы можете предлагать другим командам обменять одного из своих игроков на другого игрока.

Для каждой упорядоченной пары различных игроков (x, y) известно, согласится ли команда, управляемая компьютером, обменять вашего игрока x на своего игрока y . Определите, получится ли собрать команду, состоящую из футболистов $b_1, b_2, \dots, b_k$, и если да, выведите порядок обменов, который к этому приведет.

Формат входных данных

В первой строке даны два целых числа n и k ($1 \leq n \leq 300, 1 \leq k \leq n$) — общее количество футболистов в игре и количество футболистов в вашей команде.

Во второй строке даны k различных целых чисел a_i ($1 \leq a_i \leq n$) — текущий состав вашей команды.

В третьей строке даны k различных целых чисел b_i ($1 \leq b_i \leq n$) — желаемый состав вашей команды.

В каждой из следующих n строк записано по n символов. Символ в i -й строке и j -м столбце равен «1», если управляемая компьютером команда согласится обменять вашего игрока i на своего игрока j , и «0», если не согласится. Символы на главной диагонали равны нулю.

Формат выходных данных

В первой строке выведите «YES», если получится собрать команду из игроков $b_1, b_2, \dots, b_k$, и «NO», если не получится.

В случае ответа «YES» на второй строке выведите целое число q ($0 \leq q \leq n \cdot n$) — количество обменов, а затем q строк — последовательность обменов. Каждая из этих q строк должна содержать два различных числа x_j и y_j ($1 \leq x_j \leq n, 1 \leq y_j \leq n$) — номер игрока вашей команды, которого вы хотите обменять, и номер игрока другой команды, на которого вы хотите его обменять. Обратите внимание, что после j -го обмена игрок y_j станет игроком вашей команды, а игрок x_j перестанет быть игроком вашей команды.

Если существует несколько последовательностей обменов, ведущих к желаемому результату, разрешается вывести любую. Также обратите внимание, что не требуется минимизировать количество обменов, достаточно лишь, чтобы оно не превышало $n \cdot n$.

Примеры

стандартный ввод	стандартный вывод
5 2 1 2 4 5 00100 00100 00011 00000 00000	YES 4 1 3 3 4 2 3 3 5
3 2 1 2 2 3 000 001 010	NO

Задача Е. Переворот подстроки

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

Даны две строки s и t одинаковой длины. Требуется определить, можно ли из строки s получить строку t , ровно один раз перевернув в ней некоторую подстроку.

Формат входных данных

В первой строке записана строка s , а во второй — строка t . Обе строки имеют одинаковую длину от 1 до 200000 и состоят из строчных латинских букв.

Формат выходных данных

Выведите «YES», если можно перевернуть некоторую подстроку строки s так, чтобы s стала равна t , и «NO», если этого сделать нельзя.

Примеры

стандартный ввод	стандартный вывод
abcdefg abedcfg	YES
abcdefg abdecfg	NO

Задача F. Восстановление дерева

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

Есть дерево из n вершин. Для каждой вершины известен список всех ее потомков (не обязательно непосредственных). Требуется восстановить дерево или сказать, что такого дерева не существует.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 1000$) — количество вершин в дереве.

В каждой из следующих n строк сначала записано число c_i ($0 \leq c_i \leq n$) — количество потомков вершины i , а затем c_i различных целых чисел a_{ij} ($1 \leq a_{ij} \leq n$) — номера потомков вершины i .

Формат выходных данных

Если ответа не существует, выведите «NO».

Иначе в первой строке выведите «YES», а затем выведите $n - 1$ строку, по два числа в каждой — номер родителя и потомка. Пары (родитель, потомок) можно выводить в любом порядке.

Примеры

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

Задача G. Недопалиндромность

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

Назовем *недопалиндромностью* массива b длины k минимальное количество раз, которое надо увеличить некоторые элементы b_j на 1, так чтобы массив b в результате стал палиндромом, т.е. $b_1 = b_k$, $b_2 = b_{k-1}$, и т.д.

Дан массив длины n , состоящий из целых чисел. Рассмотрим все его подмассивы длины k , а для каждого из этих подмассивов его *недопалиндромность* p_i . Требуется вычислить сумму всех p_i ($1 \leq i \leq n - k + 1$).

Формат входных данных

В первой строке даны два числа n и k ($1 \leq k \leq n \leq 200000$) — длина массива и длина подмассивов.

Во второй строке даны n целых чисел a_i ($-10^8 \leq a_i \leq 10^8$) — элементы массива.

Формат выходных данных

Выведите единственное целое число — сумму *недопалиндромностей* всех подмассивов длины k .

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 2	3
5 3 2 3 3 1 4	4

Задача Н. Безопасный путь

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

Вы играете в новую RPG. Карта мира в ней представляет собой клетчатое поле размером $n \times m$ клеток. Любой игровой персонаж, стоящий в какой-либо клетке, может переместиться из этой клетки в четырех направлениях — в клетку слева, справа, спереди и сзади, но не выходя за карту мира.

В некоторых клетках живут монстры. Если в какой-то момент времени вы находитесь на клетке, до которой какой-либо из монстров может добраться за d шагов или меньше, он тут же прибежит и убьет вас.

Вам требуется добраться живым из одной клетки игрового поля в другую. Определить, можно ли это сделать, и если да, какое минимальное количество шагов для этого потребуется.

Формат входных данных

В первой строке даны три неотрицательных числа n , m и d ($2 \leq n \cdot m \leq 200000$, $0 \leq d \leq 200000$) — размеры поля и максимальное расстояние, на котором монстры опасны.

Каждая из n следующих строк состоит из m символов. Эти символы могут быть равны «.», «М», «S» и «F», что означает пустую клетку, клетку с монстром, стартовую клетку и финишную клетку соответственно. Стартовая и финишная клетки — пустые, и они встречаются ровно один раз.

Формат выходных данных

Если можно добраться из стартовой клетки в финишную живым, выведите минимальное количество шагов, которое для этого нужно сделать. Иначе выведите «-1».

Примеры

стандартный ввод	стандартный вывод
5 7 1 S.M...M M...M..F	12
7 6 2 S..... ...M..M M.....F	11
7 6 2 S..... ...M..M M.....F	-1
4 4 2 M... .S..F	-1

Замечание

Обратите внимание, что монстры могут прибежать и убить вас как на стартовой, так и на финишной клетке.

Задача I. Угадай дерево

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

Жюри задумало полное бинарное дерево с $n = 2^h - 1$ вершинами. Его вершины пронумерованы целыми различными числами от 1 до n , однако вам неизвестно, какие вершины с какими соединены.

Вы можете спрашивать у программы жюри, чему равно расстояние между некоторыми двумя вершинами. Вы должны с помощью не более чем $2.5 \cdot h \cdot n$ таких запросов восстановить структуру дерева.

Протокол взаимодействия

Это интерактивная задача. Ваша программа должна общаться с программой жюри, используя для этого стандартные потоки ввода и вывода.

В самом начале вашей программе сообщается единственное число n ($1 \leq n \leq 1023$, $n + 1$ — степень двойки) — количество вершин в дереве.

После этого вы можете сделать не более $2.5 \cdot h \cdot n$ запросов. Чтобы сделать запрос, выведите символ «?» и два целых числа от 1 до n — номера вершин x и y , расстояние между которыми вы хотите узнать. В ответ на такой запрос вам будет сообщено одно целое число — расстояние между вершинами x и y .

Как только вы узнаете полную структуру дерева, выведите символ «!», а затем n целых чисел p_i , где p_i — это предок вершины i , либо число 0, если вершина i является корнем дерева. После этого ваша программа должна завершиться.

Пример

стандартный ввод	стандартный вывод
<i><interactor's output></i>	3
? 1 2	<i><solution's output></i>
<i><interactor's output></i>	1
? 2 3	<i><solution's output></i>
<i><interactor's output></i>	1
! 2 0 2	<i><solution's output></i>

Замечание

Обратите внимание, что каждое выведенное вами сообщение должно завершаться переводом строки. Также после вывода каждого сообщения ваша программа должна очищать потоковый буфер, чтобы выведенная вами информация дошла до программы жюри: например, это делают вызовы «`fflush(stdout)`» или «`cout.flush()`» в C++, «`System.out.flush()`» в Java, «`Console.Out.Flush()`» в C#, «`flush(output)`» в Pascal, «`sys.stdout.flush()`» в Python.

Задача J. Параллелограммы

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

Есть n палочек, i -я из которых имеет длину a_i . Леша хочет собрать из них как можно больше параллелограммов одновременно, причем каждая палочка может быть использована не более чем в одном параллелограмме. Какое максимальное количество параллелограммов удастся собрать?

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 200000$) — количество палочек.

Во второй строке даны n чисел a_i ($1 \leq a_i \leq 200000$) — длины палочек.

Формат выходных данных

Выведите единственное целое число — максимальное количество параллелограммов, которое удастся собрать.

Примеры

стандартный ввод	стандартный вывод
4 1 2 1 2	1
12 1 3 5 7 1 3 5 7 1 3 5 7	2

Задача К. Видеообзоры

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

Студия «Lodka Gaming» занимается рекламой своей новой игры «.C.O.N.T.E.S.T: Unexpected Behaviour». Маркетолог студии планирует последовательно пообщаться с n видеоблогерами (в заданном порядке, начиная с 1-го и заканчивая n -м), предложив им записать видеообзор на игру. Все люди разные, и видеоблогеры — тоже, поэтому i -й видеоблогер будет записывать обзор в двух случаях: либо если ему интересна игра, либо если на нее уже записано не менее a_i видеообзоров.

Студия хочет, чтобы в сети появилось не менее m видеообзоров. Геймдизайнер «Lodka Gaming» понимает, что, возможно, сами собой эти видеообзоры не появятся, поэтому он хочет *убедить* некоторых видеоблогеров, что им на самом деле интересна эта игра. Какое минимальное количество видеоблогеров нужно *убедить*?

Формат входных данных

В первой строке даны два целых числа n и m ($1 \leq n \leq 200000, 1 \leq m \leq n$) — количество видеоблогеров и требуемое количество видеообзоров.

Во второй строке даны n целых чисел a_i ($0 \leq a_i \leq 200000$) — минимальное количество видеообзоров, которые должны появиться в сети, чтобы i -й видеоблогер записал обзор в случае, если ему неинтересна игра.

Формат выходных данных

Выведите единственное число — минимальное количество видеоблогеров, которых придётся *убедить* записать обзор на игру, чтобы в сети появилось не менее m обзоров.

Примеры

стандартный ввод	стандартный вывод
7 4 2 1 3 3 4 2 3	1
7 4 2 1 3 3 4 3 2	2

Задача L. Запросы на строке

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

Дана строка s . Также есть строка p , и вначале она пустая. Вам требуется выполнить q операций вида «добавить букву в конец строки p » и «удалить букву с конца строки p », а после выполнения каждой операции нужно сказать, содержится ли p в s в качестве подпоследовательности.

Формат входных данных

В первой строке дана строка s , имеющая длину от 1 до 200000 и состоящая из строчных латинских букв.

Во второй строке дано целое число q ($1 \leq q \leq 200000$) — количество операций.

В каждой из следующих q строк дано описание операции в формате «push c », что означает «добавить букву c в конец строки p » (c — строчная латинская буква), или «pop», что означает «удалить букву с конца строки p ». Гарантируется, что операция «pop» никогда не применяется к пустой строке p .

Формат выходных данных

Выведите q строк, каждая из которых равна «YES» или «NO», в зависимости от того, содержится ли строка p в строке s в качестве подпоследовательности после выполнения очередной операции.

Пример

стандартный ввод	стандартный вывод
abccabc	YES
30	YES
push a	YES
pop	YES
push a	NO
push a	YES
push a	YES
pop	NO
push c	YES
push b	YES
pop	YES
pop	YES
push b	NO
push c	YES
push c	YES
pop	YES
pop	YES
pop	YES
pop	YES
push b	YES
push c	YES
push c	YES
pop	YES
push b	YES
push c	YES
pop	YES
pop	YES
push a	YES
push b	NO
push c	YES
push a	
pop	

Задача М. Забытое заклинание

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

Три мага пытаются вспомнить заклинание. Каждый из них считает, что это некоторая строка длины n , и известно, что каждый из них ошибается не более чем в одной букве. Требуется восстановить забытое заклинание.

Формат входных данных

Во входных данных даны три строки одинаковой длины n ($1 \leq n \leq 200000$), состоящие из строчных латинских букв.

Формат выходных данных

Если такая ситуация невозможна, выведите «Impossible».

Если существует несколько заклинаний, удовлетворяющих условию, выведите «Ambiguous».

Наконец, если забытое заклинание единственно возможное, выведите его.

Примеры

стандартный ввод	стандартный вывод
aab aca daa	aaa
abc aca abc	Ambiguous
abcde fghij klmno	Impossible