

"А" Перевод времени

Ограничение по времени 3 секунды

Ограничение по памяти 256 мегабайт

В стране Флатландии n городов. Флатландия находится не на Земле, а потому сутки во Флатландии делятся $2s$ часов, а в году T суток, то есть ровно $2sT$ часов.

Каждый город Флатландии имеет свое правительство, и так случилось, что в каждом городе приняты свои законы о переводе часов. В некоторых городах, к примеру, время бывает не только летнее и зимнее, но и весеннее, осеннее, демисезонное, праздничное и выходное. Так как законы в каждом городе свои, то часто случается ситуация, когда время в разных городах сильно отличается. Естественно, наличие большого числа различных времен вызывает неудобство у жителей страны.

Город, имеющий номер n , является столицей Флатландии. В нем, ради удобства общения с другими странами, время не переводят никогда. Также известно, что разница между временем любого другого города и временем столицы никогда не превышает s часов по абсолютной величине.

Опишем, как происходит процесс измерения времени и перевода часов. В центре каждого города стоит огромные часы. Эти часы показывают номер текущего дня в этом городе, а также текущий час. Минуты и секунды на часах не показываются, так как правительство считает их неважными. В начале года, по старой традиции, время во всех городах синхронизируется, и год в каждом городе начинается в полночь, то есть все часы показывают ноль и начинается первый день. При каждом наступлении полуночи в городе начинается новый день, то есть меняется номер текущего дня в этом городе. Если в городе наступает день перевода, то при первом наступлении полудня (часы показывают s часов) этого дня происходит перевод часов на указанную в законе величину. Перевод никогда не меняет номер текущего дня, однако если перевод производится на s часов вперед, то сразу же наступает полночь и, соответственно, следующий день в этом городе. Специальная Комиссия по Оценке Неудобства Флатландии разработала численную характеристику, позволяющую оценить возникающее неудобство. Часовым неудобством в стране называется величина, равная сумме модулей разниц времен для всех пар городов в течении некоторого часа. А именно, обозначим как ti величину, равную $2sdi + ci$, где di — номер текущего дня в i -ом городе, а ci — номер часа в i -ом городе. Просуммируем величину $|ti - tj|$ для всех неупорядоченных пар различных городов $\{i, j\}$. Полученная сумма и есть часовое неудобство в стране. Неудобством за год является сумма часовых неудобств по всем часам в году, то есть по $2sT$ часам с начала года. Известны графики перевода времен в городах в течение года. Ваша задача состоит в том, чтобы посчитать неудобство за год. В начале года времена по всех городах совпадают и год начинается с полуночи, то есть на часах ноль. Перевод времени в любом городе происходит только в полдень по местному времени, то есть в s часов. Переводы, связанные с синхронизацией часов в начале года не указаны, так как они являются крупным культурным событием и происходят все одновременно с наступлением нового года.

Формат входных данных

Первая строка содержит четыре целых числа n , m , s и T ($2 \leq n \leq 104$; $1 \leq m \leq 105$; $1 \leq s \leq 104$; $1 \leq T \leq 100$). Далее следуют m строк, описывающих переводы времени. Каждое описание состоит из трех чисел di , ki и ti — номер дня, в который происходит перевод времени, номер города, в котором происходит перевод, и число часов, на которые переводится время ($1 \leq di$; $1 \leq ki \leq n - 1$; $-s \leq ti \leq s$; $ti \neq 0$). Каждый перевод совершается не более, чем на 200 часов.

Гарантируется, что данные во входном файле корректны. В каждом городе происходит не

более одного перевода в день. Дни в году Флатландцы нумеруют с единицы.

Формат выходных данных

Выведите единственное целое число — неудобство за год. Гарантируется, что ответ не превысит $8 \cdot 10^{18}$.

Входные данные

4 4 12 3

1 1 1

2 2 1

3 1 -1

3 2 -1

Выходные данные

164

"В" Обратная задача о черепашке
Ограничение по времени 3 секунды
Ограничение по памяти 256 мегабайт

Одной из задач, решаемых методом динамического программирования, является задача о черепашке. Приведем условие этой задачи.

Задано поле размером n на m клеток. При этом для каждой клетки поля задано, является она *разрешенной* или *запрещенной* (левая верхняя и правая нижняя клетки обязательно разрешены). В левой верхней клетке находится черепашка. Черепашка за один ход может переместиться в соседнюю разрешенную клетку поля, которая находится либо справа, либо снизу от клетки, в которой она сейчас находится. Необходимо определить число путей, которыми черепашка может достичь правой нижней клетки поля, начав путь из левой верхней клетки.

Вам предлагается решить обратную задачу, а именно, найти такое поле, что число путей черепашки будет равно k .

Формат входных данных

Первая строка содержит число k ($1 \leq k \leq 10^{18}$).

Формат выходных данных

В первой строке выведите два целых числа, разделенных пробелом, — n и m . Эти числа не должны быть менее единицы или превышать 300.

Затем выведите n строк по m символов в каждой — описание поля. Разрешенной клетке поля должен соответствовать символ «1», запрещенной — «0». Левая верхняя и правая нижняя клетки поля должны быть разрешенными.

При наличии нескольких полей, удовлетворяющих указанным требованиям, выведите любое из них. Существование хотя бы одного такого поля гарантируется.

Входные данные

2

Выходные данные

2 3

111

011

"С" Разбиения таблицы

Ограничение по времени 3 секунды

Ограничение по памяти 256 мегабайт

Рассмотрим числовую таблицу $A[1..n, 1..m]$, заполненную нулями и единицами. Любая четверка целых чисел (r_1, r_2, c_1, c_2) , такая что $1 \leq r_1 < r_2 < n$ и $1 \leq c_1 < c_2 < m$, задает разбиение таблицы на девять частей, показанное на рисунке.

	1	c_1		c_2		m
1	$A_1 = A[1..r_1, 1..c_1]$		$A_2 = A[1..r_1, c_1+1..c_2]$		$A_3 = A[1..r_1, c_2+1..m]$	
r_1						
	$A_4 = A[r_1+1..r_2, 1..c_1]$		$A_5 = A[r_1+1..r_2, c_1+1..c_2]$		$A_6 = A[r_1+1..r_2, c_2+1..m]$	
r_2						
	$A_7 = A[r_2+1..n, 1..c_1]$		$A_8 = A[r_2+1..n, c_1+1..c_2]$		$A_9 = A[r_2+1..n, c_2+1..m]$	
n						

Рисунок. Разбиение таблицы A на девять частей

Обозначим как $sum(A_i)$ сумму чисел в части A_i . Пусть $S = sum(A_1) + sum(A_3) + sum(A_5) + sum(A_7) + sum(A_9)$. Ваша задача — определить для заданной таблицы A число таких разбиений, что S четно.

Например, для таблицы

0101

0101

0100

существует три разбиения. При $(r_1=1, r_2=2, c_1=1, c_2=2)$ и $(r_1=1, r_2=2, c_1=1, c_2=3)$ разбиения таковы, что сумма чисел в частях с нечетными номерами равна двум, то есть четна. При $(r_1=1, r_2=2, c_1=2, c_2=3)$ сумма равна трем, то есть нечетна. Таким образом, подходит два разбиения.

Формат входных данных

Первая строка содержит два целых числа n и m ($3 \leq n, m \leq 3000$). Каждая из следующих n строк содержит по m символов в каждой — описание таблицы A .

Формат выходных данных

Выведите число четверок чисел (r_1, r_2, c_1, c_2) , таких, что:

- $1 \leq r_1 < r_2 < n$;
- $1 \leq c_1 < c_2 < m$;
- сумма чисел в частях таблицы A с нечетными номерами четна.

Входные данные

3 3

110

101

010

Выходные данные

0

Входные данные

3 4

0101

0101

0100

Выходные данные

2

"D" Архиватор

Ограничение по времени 3 секунды

Ограничение по памяти 256 мегабайт

Вася придумал радикально новый способ сжатия текста. Он считает, что особенно хорошо его метод будет работать для языков с большим алфавитом, таких как китайский. По этой причине Вася представляет текст в виде последовательности кодов символов $a_1, a_2, \dots, a_n$.

Алгоритм сжатия разбит на фазы. На каждой фазе строится новая последовательность вдвое меньшей длины $b_1, b_2, \dots, b_{\lceil n/2 \rceil}$. При этом b_i может равняться либо a_i , либо a_{n-i+1} . Выбор оставляется на усмотрение архиватора. После этого исходная последовательность заменяется новой, и алгоритм переходит к следующей фазе. Процесс завершается, когда длина последовательности становится равной единице. Вася заметил, что в случае неоднозначности ($a_i \neq a_{n-i+1}$) приходится сохранять дополнительные данные во избежание потери информации. Кроме того, от выбора символа на текущей фазе может зависеть наличие или отсутствие неоднозначностей в дальнейшем.

Для оценки эффективности алгоритма Вася решил назначать единицу штрафа за каждую неоднозначность, возникающую в процессе работы архиватора. К сожалению, Вася не смог придумать, каким образом нужно делать выбор при возникновении неоднозначностей, и обратился к вам за помощью. Он просит вас написать программу, которая вычислит минимальный возможный суммарный штраф при оптимальном разрешении неоднозначностей.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 200000$). Вторая строка содержит исходную последовательность — n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите целое число — минимальный суммарный штраф, который может быть достигнут при сжатии заданной последовательности по описанному алгоритму.

Входные данные

5

1 1 2 1 2

Выходные данные

2

"Е" Блэкджон

Ограничение по времени 3 секунды

Ограничение по памяти 256 мегабайт

С недавних пор блэкджек не приносит Джону былого удовольствия. Все вероятности давно просчитаны, оптимальная стратегия игры выработана. Поэтому Джон решил создать свою собственную модификацию этой игры — блэкджон.

Джон взял колоду карт и на каждой написал дробь p/q , по модулю не превосходящую единицу. Эта дробь называется значением карты. Правила таковы: игрок по очереди открывает карты из колоды и каждую либо берет в руку, либо сбрасывает. Для выигрыша необходимо добиться того, чтобы сумма значений карт в руке стала равной единице. Джон хочет продемонстрировать новую игру друзьям, но у него есть серьезная проблема. Игра получилась настолько сложная, что даже зная заранее колоду карт, он не может определить, какие карты надо брать.

Джон попросил вас написать программу, которая по заданной колоде карт определит, какие карты надо брать в руку для того, чтобы выиграть.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 100$) — количество карт в колоде. Каждая из следующих n строк описывает одну из карт и содержит два разделенных пробелом целых числа p_i и q_i ($1 \leq q_i \leq 21$, $|p_i| \leq q_i$).

Формат выходных данных

В случае, если выиграть невозможно, выведите единственную строку NO.

В противном случае в первой строке выведите YES. Во второй строке выведите целое число m — количество карт в руке к концу игры. В третьей строке выведите m чисел $k_1, k_2, \dots, k_m$ — номера карт, сумма значений которых равна единице.

Карты нумеруются с единицы в порядке их перечисления во входном файле.

Входные данные

4

1 2

-1 6

1 5

2 3

Выходные данные

YES

3

1 2 4

Входные данные

4

1 2

1 4

1 8

1 16

Выходные данные

NO

"F" Разрезание торта

Ограничение по времени 3 секунды

Ограничение по памяти 256 мегабайт

Паше на день рождения подарили торт. На этом торте есть два типа украшений — розочки из крема и вишенки. Паша хочет отрезать себе кусочек торта таким образом, чтобы на нем была хотя бы одна розочка из крема, но не было ни одной вишенки. Если смотреть на торт сверху, то его можно представить как квадрат. Если ввести прямоугольную декартову систему координат с началом координат в центре торта и осями координат, параллельными сторонам торта, то координаты одного из углов торта будут равны $(-106, -106)$, а противоположного угла — $(106, 106)$. При этом все розочки из крема и вишенки находятся строго внутри этого квадрата.

Для того чтобы отрезать себе кусочек торта, Паша хочет провести один прямолинейный разрез. При этом, чтобы разрезать было проще, он хочет, чтобы угол между прямой, по которой он будет резать, и осью Ox был как можно меньше.

Ваша задача состоит в том, чтобы определить, может ли Паша разрезать торт описанным способом, и найти минимальный возможный угол между линией разреза и осью Ox .

Если розочка из крема или вишенка находятся на линии разреза, то можно считать, что Паша так проводит разрез, что розочка или вишенка окажутся по ту сторону от разреза, по какую необходимо.

На рисунке штрих-пунктирной линией показан оптимальный способ разрезания торта для первого примера. Точки, в которых расположены розочки, обозначены на рисунке красным цветом. Точки, в которых расположены вишенки, обозначены синим цветом.

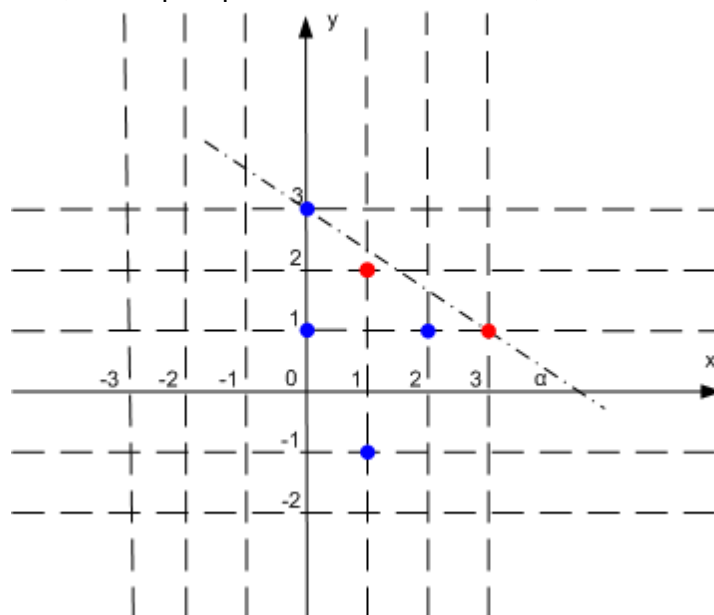


Рисунок. Расположение розочек и вишенок для первого примера

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 100000$) — количество розочек из крема. Каждая из последующих n строк содержит по два целых числа — координаты одной из розочек.

Далее следует строка, содержащая целое число m ($1 \leq m \leq 100000$) — количество вишенок. Каждая из последующих m строк содержит по два целых числа — координаты одной из вишенок.

Никакие две точки, в которых находятся розочки из крема и вишенки, не совпадают. Координаты розочек и вишенок строго меньше 10^6 по абсолютной величине.

Формат выходных данных

Если отрезать кусочек торта указанным способом невозможно, выведите в выходной файл слово Impossible.

Иначе, выведите в выходной файл минимальный угол между осью Oх и прямой, описывающей разрез. Угол необходимо вывести в радианах, с точностью не хуже 10^{-4} .

Входные данные

2

1 2

3 1

4

0 1

2 1

0 3

1 -1

Выходные данные

0.5880026035475675

Входные данные

1

1 1

2

0 1

2 1

Выходные данные

0

Входные данные

1

10 10

4

10 11

10 9

11 10

9 10

Выходные данные

Impossible