

Задача А. Битва за пульт

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

Сдаётся мне, джентльмены, это была... комедия.

— Одноглазый ковбой

Петя и Маша хорошие друзья. Этим вечером они решили включить телевизор. Петя предпочитает смотреть боевики. Маша - комедии. Каждый раз у них происходит битва за пульт, чтобы переключить канал у телевизора. Формально вам дано n фильмов. Каждый фильм характеризуется временем начала и временем конца, а также у него есть тип:

1. боевик;
2. комедия;
3. боевик и комедия.

Фильмы необязательно смотреть от начала до конца. Можно смотреть произвольные фрагменты и в любой момент переключаться между разными фильмами. Счастье каждого друга определяется как количество просмотренных им минут фильма его любимого жанра. Когда друзья смотрят фильм с типом 3, то каждый из них счастлив в тот промежуток времени. Вам нужно определить суммарное максимальное счастье друзей при условии, что в каждый момент времени можно смотреть только один фильм.

Формат входных данных

Первая строка содержит число фильмов n ($1 \leq n \leq 10^5$). В следующих n строках задается l_i, r_i ($1 \leq l_i < r_i \leq 10^9$), $t = 1, 2, 3$. Это время начала, окончания фильма и его жанр. В l минуту фильм начинается и до начала r минуты фильм заканчивается.

Формат выходных данных

Выведите одно число — максимальную сумму счастья обоих друзей.

Примеры

стандартный ввод	стандартный вывод
5 1 2 1 2 3 2 4 5 3 5 6 1 10 20 1	15
3 1 15 2 2 5 3 4 7 3	19

Задача В. Замощение трапециями

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

Вы заставляете эту проблему выходить за пределы измерений больше, чем она есть на самом деле.

— Безумный ученый.

У Максима есть две прямоугольные трапеции в двух виртуальных мирах. Он решил их расположить в новом мире и провести геометрический эксперимент. Затем он задался вопросом: можно ли расположить трапеции так, чтобы из них получился один прямоугольник. Разрешается поворачивать трапеции на любой угол и перемещать их в любое место, при этом накладывать трапеции друг на друга нельзя.

Вам даны координаты трапеций в порядке обхода по часовой стрелке, начиная с нижнего левого угла. Гарантируется, что изначально основания обеих трапеций параллельны оси X и нижний левый угол всегда прямой. Помогите Максиму ответить на его вопрос.

Формат входных данных

Первая строка содержит 4 пары точек $x_{11}, y_{11}, x_{12}, y_{12}, x_{13}, y_{13}, x_{14}, y_{14}$ первой трапеции.

Вторая строка содержит 4 пары точек $x_{21}, y_{21}, x_{22}, y_{22}, x_{23}, y_{23}, x_{24}, y_{24}$ второй трапеции.

Гарантируется, что точки образуют трапецию, её основания параллельны оси X , нижний угол всегда прямой и каждая из трапеций является прямоугольной, то есть, имеет **ровно** два прямых угла и не вырождается в треугольник. Все координаты являются целочисленными, неотрицательными и не превосходят $3 \cdot 10^4$.

Формат выходных данных

Выведите «YES» без кавычек, если можно из трапеций создать прямоугольник и «NO» без кавычек в противном случае.

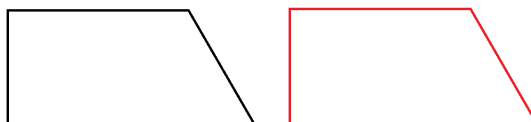
Примеры

стандартный ввод	стандартный вывод
1 1 1 3 4 3 5 1 6 1 6 3 9 3 10 1	YES
1 1 1 2 4 2 3 1 0 0 0 1 5 1 3 0	NO

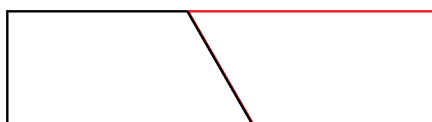
Замечание

На рисунке показан первый тест.

Изначальное размещение трапеций:



Позиция трапеций после соединения:



Задача С. Что-то про последовательность

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

*После хорошего обеда я люблю лежать
на полу и считать k -й элемент
последовательности*

— Призрачный музыкант

У Максима есть последовательность натуральных чисел от 1 до 10^{18} . Максим мог всегда уверенно сказать, какое число стоит на k -й позиции в этой последовательности.

Но Никита решил, что это слишком легко, и удалил несколько чисел из этой последовательности. Более формально, Никита совершил x раз следующую операцию: он одновременно удалил из последовательности все числа на позициях $y, 2 \cdot y, 3 \cdot y, \dots, q \cdot y \leq |seq|$, где $|seq|$ — длина текущей последовательности.

После того, как Максим увидел что осталось от его последовательности, он пришел в замешательство. Однако ему по-прежнему хочется уметь определять, какое число стоит на k -й позиции в получившейся последовательности, и вообще, существует ли теперь k -й элемент, ведь последовательность могла стать короче.

Помогите Максиму найти ответ на этот вопрос.

Давайте рассмотрим пример. Пусть $x = 2, y = 3, k = 5$:

1 2 3 4 5 6 7 8 9 10 11 12 ...
1 2 ~~3~~ 4 5 ~~6~~ 7 8 ~~9~~ 10 11 ~~12~~ ...
1 2 ~~3~~ ~~4~~ 5 ~~6~~ 7 ~~8~~ ~~9~~ 10 11 ~~12~~ ...

Числа, зачеркнутые красной линией, удалены после первой операции Никиты, а числа, зачеркнутые синей линией, удалены после второй операции. Тогда число на $k = 5$ позиции стоит число 10.

Формат входных данных

Единственная строка содержит 3 целых числа x, y, k ($1 \leq y, k \leq 10^{18}, 1 \leq x \leq 10^5$).

Формат выходных данных

Выведите одно целое число — k -й элемент в получившейся последовательности, или -1 , если k -го элемента не существует.

Примеры

стандартный ввод	стандартный вывод
2 3 5	10
2 5 1	1
20 2 10000000000000000	-1

Задача D. Xor-изация

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	5 секунд
Ограничение по памяти:	256 мегабайт

*Bury the light deep within!
Cast aside, there's no coming home!
We're burning chaos in the wind!
Drifting in the ocean all alone!*

Каждый раз, когда Максим едет в поезде, он придумывает новую задачу. В этот раз формулировка задачи следующая:

Изначально есть массив a , состоящий из n целых неотрицательных чисел. Также есть q запросов для этого массива, каждый одного из двух типов:

- 1 l r x — ко всем числам a_i на отрезке от l до r применяется операция исключающего ИЛИ* с числом x , то есть $a_i := a_i \oplus x$ для всех i таких, что $l \leq i \leq r$.
- 2 l r — посчитать минимальное значение $a_i \oplus a_j$ по всем парам i, j таких, что $l \leq i < j \leq r$.

Сможете ли вы решить задачу Максима?

*Побитовое исключающее ИЛИ — бинарная операция, действие которой эквивалентно применению логического исключающего ИЛИ к каждой паре битов, которые стоят на одинаковых позициях в двоичных представлениях операндов. Другими словами, если оба соответствующих бита операндов равны между собой, двоичный разряд результата равен 0; в противном случае, двоичный разряд результата равен 1.

Формат входных данных

Первая строка содержит два целых числа n ($2 \leq n \leq 10^5$) и q ($1 \leq q \leq 10^5$).

Вторая строка содержит n целых чисел a_i ($0 \leq a_i \leq 1000$).

Далее следуют q строк с описанием запросов.

Для каждого запроса ($1 \leq l \leq r \leq n$), ($0 \leq x \leq 1000$). Также для каждого запроса типа 2 гарантируется, что $l < r$.

Формат выходных данных

Для каждого запроса типа 2 выведите одно целое неотрицательное число — ответ на запрос.

Примеры

стандартный ввод	стандартный вывод
5 3 8 2 5 1 7 2 1 5 1 2 4 3 2 1 5	2 1
5 3 1 5 0 0 3 1 1 1 3 2 1 2 2 2 5	7 0
12 20 123 61 102 15 47 48 4 72 102 33 17 56 1 7 8 85 2 7 12 2 3 9 2 4 7 2 3 12 1 6 12 38 2 5 6 2 2 8 1 2 4 14 2 6 9 1 3 9 52 2 1 7 1 3 10 58 2 5 6 2 9 12 1 5 11 115 1 5 9 113 2 5 7 2 3 7 2 6 12	12 0 31 0 57 6 45 6 57 10 57 21 2

Задача Е. 21 очко

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

Вы что, в киосках их заряжаете?

— Любопытный игрок

Никита играет в 21 очко. Игра ведётся по классическим правилам: имеется стандартная колода в 52 карты, для победы игроку необходимо составить такой набор карт, что в сумме они дадут 21 очко. Каждая карта имеет свой вес:

1. 2-10 по значению;
2. J (валет) - 2;
3. Q (дама) - 3;
4. K (король) - 4;
5. A (туз) - 11;

На текущий момент игры у Никиты есть n карт. Он возьмет одну случайную карту из оставшихся $52 - n$ карт. При этом он хочет знать вероятность того, что после этого хода у него станет ровно 21 очко. Гарантируется, что изначальная колода карт была правильной - все карты различны и не может быть больше четырёх карт одного значения.

Формат входных данных

Первая строка содержит количество карт n ($1 \leq n \leq 52$). В следующих n строках задаются значения этих карт: 2-10, J , Q , K , A . Гарантируется, что в руке Никиты нет 5 карт одинакового значения.

Формат выходных данных

Выведите одно число, вероятность того, что следующим ходом Никита наберёт 21 очко. Решение будет засчитано, если ответ участника отличается от ответа жюри не более, чем на 10^{-6} .

Примеры

стандартный ввод	стандартный вывод
3 5 Q 8	0.0612244898
8 4 2 2 2 J J J 3	0.0454545455

Замечание

В первом примере. Никита набрал 16 очков. Ему нужно ещё 5. Так как одна пятёрка уже есть в руке, то остаётся всего три варианта. Ответ $\frac{3}{49}$.

Задача F. Но вы обо мне слышали

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

Открывать миллиард сундуков не входило в план.

— Жалкий пират

Максим неожиданно стал пиратом! И как и полагается пирату, Максим отправился искать сокровища.

Ему улыбнулась удача, и на каком-то неизвестном острове он обнаруживает n сундуков. Максим знает, что только один сундук набит золотом и драгоценными камнями, а в остальных лежит барахло. И вот, выставив в ряд все найденные сундуки и пронумеровав их от 1 до n , наш новоиспеченный пират планирует их вскрыть. Он будет открывать их последовательно либо с начала (т.е. откроет 1, потом 2, потом 3 и т.д.), либо с конца (т.е. откроет n , потом $n - 1$, потом $n - 2$ и т.д.) до тех пор, пока не найдет золото и драгоценности. Максим не знает, в каком именно порядке лучше это делать, поэтому просит у вас помощи.

Если предположить, что драгоценности на самом деле находятся в сундуке под номером k , то сколько сундуков придется открыть не знающему это Максиму в худшем случае?

Формат входных данных

Единственная строка содержит два целых числа n, k ($1 \leq k \leq n \leq 10^9$).

Формат выходных данных

Выведите одно целое число – ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
1 1	1
5 4	4
10 3	8
999999999 465295903	534704097

Задача G. Дискриминант или теорема Виета?

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

*Не знаю, как это спасет мир, но
подождите, я занят этим всего 1
секунду.*

— Гениальный детектив.

Никита записал на доске квадратное уравнение в виде:

$$a \cdot x^2 + b \cdot x + c = 0$$

При этом коэффициенты a , b , c **ненулевые** и **целые** числа.

Никите не очень понравилось данное уравнение, поэтому он захотел изменить минимальное количество коэффициентов на другие (тоже **ненулевые** и **целые**) так, чтобы после этого квадратное уравнение имело два совпадающих действительных корня.

Помогите Никите заменить минимальное количество коэффициентов!

Формат входных данных

В единственной строке содержится три целых числа a , b , c ($-10^4 \leq a, b, c \leq 10^4, a \neq 0, b \neq 0, c \neq 0$).

Формат выходных данных

Выведите три целых числа a' , b' , c' ($-10^9 \leq a', b', c' \leq 10^9, a' \neq 0, b' \neq 0, c' \neq 0$) – новые коэффициенты уравнения.

Примеры

стандартный ввод	стандартный вывод
1 2 3	1 2 1
1197 -144 3325	1197 3990 3325
-3 6 -3	-3 6 -3
-2 5 3	-2 -20 -50

Замечание

В первом примере заменили 1 коэффициент c .

Во втором примере заменили 1 коэффициент b .

Во третьем примере заменили 0 коэффициентов.

В четвертом примере заменили 2 коэффициента b и c .

Во всех примерах можно доказать, что нельзя заменить меньшее количество коэффициентов так, чтобы получить квадратное уравнение с двумя совпадающими действительными корнями.

Задача Н. Халявка

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

*Его не подкупить, с ним не договориться.
Ему неизвестна жалость, раскаяние или
страх. Его не остановит ничто, пока он
не уберет комнату.*

— Продавец в магазине техники.

У Максима есть умный робот-пылесос. Робот сделан с помощью новейших японских технологий, поэтому работает он следующим образом:

Каждая комната, в которой он работает представляется в виде квадрата, разделенного на n^2 одинаковых областей*. После того, как робот убирается в какой то одной области, он переезжает в соседнюю по стороне* и начинает убираться там, при этом робот никогда не посещает одну и ту же область дважды.

Для того, чтобы робот работал корректно, в его память нужно загрузить таблицу $n \times n$ с приоритетами областей. С её помощью умная машина строит свой маршрут таким нехитрым алгоритмом:

Пока текущая область не (n, n) :

Убрать текущую область.

Если есть хотя бы одна не посещенная соседняя по стороне область:

Перейти в еще не посещенную соседнюю по стороне область с наивысшим приоритетом.

Иначе:

Взорваться.

Убрать текущую область.

Завершить работу.

*Область (x, y) — x -я сверху, y -я слева. То есть $(1, 1)$ это левая верхняя, а (n, n) это нижняя правая области.

*Соседние по стороне области — это такие области, у которых одна координата различается ровно на 1, а вторая координата совпадает.

Естественно, у такого алгоритма очень много недостатков, поэтому нужно строго придерживаться правил составления таблицы, а именно:

- Каждая из n^2 клеток таблицы имеет свой приоритет, который задается латинской буквой от 'a' до 'z' ('a' - наивысший приоритет, 'b' - меньший приоритет, и так далее до 'z').
- В любой момент времени робот должен однозначно определять область, в которую он переезжает (то есть из всех еще не посещенных соседних по стороне областей должна быть только одна с максимальным приоритетом), иначе он просто сломается.

Максим хочет, чтобы робот убрал **максимальную площадь комнаты (то есть максимальное количество областей)**. Помогите Максиму составить таблицу приоритетов по всем правилам такую, что робот начнет уборку из области $(1, 1)$, уберет максимальную площадь комнаты и при этом не сломается и не взорвется во время уборки, то есть закончит в области (n, n) .

Формат входных данных

В первой строке содержится единственное целое число n ($2 \leq n \leq 26$).

Формат выходных данных

Выведите n строк размера n из символов от 'a' до 'z' - таблицу для робота.

Примеры

стандартный ввод	стандартный вывод
2	ab aa
3	bxz bhz eez
4	abcd hgfe hixx hiqq

Замечание

Обратите внимание, что если робот посетил область (n, n) , то ни в какие другие области он уже не сможет попасть, так как после того, как робот уберется в (n, n) , он мгновенно завершит работу.

Задача I. Деление строки

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

Мы делили, делили и наконец поделили.

— Сказочный персонаж

У Максима есть строка s длины n . Он хочет разделить её на k непустых частей (не обязательно одинаковой длины) так, что в каждой получившейся строке будет одинаковое количество согласных букв.

Напомним, что гласными в английском алфавите считаются следующие буквы: 'a', 'e', 'i', 'o', 'u', 'y'. Все остальные буквы - согласные.

Сможет ли Максим разделить строку таким образом?

Формат входных данных

В первой строке содержатся два целых числа n и k ($1 \leq n, k \leq 2 \cdot 10^5$). Вторая строка содержит s длиной n символов, состоящую только из маленьких латинских букв.

Формат выходных данных

Выведите «Yes» или «No» без кавычек - ответ на задачу.

Вы можете выводить «Yes» и «No» в любом регистре (например, строки «yEs», «yes» и «Yes» будут распознаны как правильный ответ).

Примеры

стандартный ввод	стандартный вывод
1 1 a	Yes
1 1 c	Yes
9 2 polnareff	Yes
6 3 jotaro	Yes
3 4 dio	No

Замечание

Давайте рассмотрим третий тест.

Исходную строку «polnareff» можно разделить на 2 части: «polna» и «reff». В обеих частях по 3 согласные буквы.

Задача J. Скобка, скобка, скобка...

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

Не волнуйся, голова! Теперь будет думать компьютер.

— Работник атомной станции

Однажды вечером, после очередной пары по дискретной математике, Максим обнаружил в левом кармане скобочную последовательность* s , а в правом кармане скобочную последовательность t . Максим решил, что так много ему не нужно, поэтому он захотел объединить их в одну **правильную** скобочную последовательность* p . Затем Максим задумался, вдруг ему еще понадобятся отдельно s и t и стал строить p по таким правилам:

- Каждый символ в p будет однозначно соответствовать символу либо из s , либо из t .
- Порядок следования символов, которые были в s , в p будет таким же как в s .
- Порядок следования символов, которые были в t , в p будет таким же как в t .

Пример p для второго теста

Помогите Максиму объединить s и t в p , или скажите, что это невозможно.

*Скобочная последовательность — символьная последовательность, составленная в алфавите, состоящем из символов ‘(’ и ‘)’’.

*Правильная скобочная последовательность (ПСП) — скобочная последовательность, удовлетворяющая следующим правилам:

- пустая строка — ПСП;
- '(' + ПСП + ')' — ПСП;
- ПСП + ПСП тоже ПСП.

Формат входных данных

В первой строке содержится s ($1 \leq |s| \leq 2000$).

Во второй строке содержится t ($1 \leq |t| \leq 2000$).

Формат выходных данных

Если можно объединить эти две скобочные последовательности в одну правильную скобочную последовательность p , то в первой строке выведите «YES» без кавычек, а во второй строке p . Если существует несколько подходящих ответов, то выведите любой.

В противном случае выведите в единственной строке «NO» без кавычек.

Примеры

стандартный ввод	стандартный вывод
((()())() (() (()))	YES (() (()) (()) (()))
(()))(((()	YES (() (()) (()))
) (	YES ()
)))((((()	NO