

Задача А. Максимальная прочность

Имя входного файла:	стандартный ввод или input.txt
Имя выходного файла:	стандартный вывод или output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Федя играет в новую игру «The Legend of Link», в которой одной из способностей персонажа является монтаж **двух** материалов в одно оружие. Каждый материал имеет свою прочность, которую можно обозначить целым положительным числом. Прочность полученного оружия определяется как сумма модулей разностей цифр в **десятичной** записи чисел на каждой позиции у двух материалов, выбранных для монтажа.

Формально, пусть первый материал имеет прочность $X = \overline{x_1x_2\dots x_n}$, а второй имеет прочность $Y = \overline{y_1y_2\dots y_n}$. Тогда прочность оружия вычисляется как $|x_1 - y_1| + |x_2 - y_2| + \dots + |x_n - y_n|$. Если числа имеют различные длины, то более короткое число **дополняется ведущими нулями**.

Федя имеет в своем инвентаре в неограниченном количестве материалы со всеми возможными прочностями от L до R включительно, помогите ему найти максимально возможную прочность оружия, которую он может получить, выбрав ровно 2 материала (возможно одинаковых) для монтажа.

Запись $C = \overline{c_1c_2\dots c_k}$ определяет число, полученное последовательной записью десятичных цифр c_1, c_2, \dots, c_k слева направо, т.е. $C = 10^{k-1} \cdot c_1 + 10^{k-2} \cdot c_2 + \dots + c_k$.

Формат входных данных

В первой строке вводится число L ($1 \leq L < 10^{10^5}$) — десятичная запись числа, обозначающая минимальную прочность материалов.

Во второй строке вводится число R ($L \leq R < 10^{10^5}$) — десятичная запись числа, обозначающая максимальную прочность материалов.

Гарантируется, что числа L и R не содержат ведущих нулей.

Обратите внимание, что входные данные могут не помещаться в стандартные 32-битные или 64-битные целочисленные типы данных.

Формат выходных данных

Выведите одно целое число — максимально возможную прочность оружия, которую может получить Федя из данных материалов.

Примеры

стандартный ввод	стандартный вывод
53 57	4
179 239	19
132228 132228	0
54943329752812629795 55157581939688863366	163
88 1914	28

Замечание

В первом примере оружие, сделанное из материалов с характеристиками 53 и 57, будут иметь наибольшую прочность: $|5 - 5| + |3 - 7| = 4$.

Во втором примере максимальная прочность достигается при материалах с характеристиками 190 и 209: $|1 - 2| + |9 - 0| + |0 - 9| = 19$.

В третьем примере есть только одна допустимая прочность, поэтому ответ 0.

В пятом примере максимальная прочность достигается при материалах с прочностями 1909 и 90: $|1 - 0| + |9 - 0| + |0 - 9| + |9 - 0| = 28$. Обратите внимание, что более короткое число было дополнено ведущими нулями.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $R - L \leq 500$ и $R \leq 10^{36}$, наберут не менее 40 баллов.

Решения, корректно работающие, когда первые цифры в L и R различны, наберут не менее 20 баллов.

Решения, корректно работающие, когда количество цифр в L строго меньше, чем количество цифр в R , наберут не менее 20 баллов.

Дополнительно, решения, корректно работающие при $R \leq 10^{18}$ будут набирать не менее половины от указанных выше баллов.

Обратите внимание, некоторые тесты могут соответствовать сразу нескольким дополнительным ограничениям.

Задача В. Игра с переворотом

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Алиса и Боб играют в игру. У них есть две строки S и T одинаковой длины n , состоящие из маленьких латинских букв. Игроки ходят по очереди, первой ходит Алиса.

В свой ход Алиса выбирает число i от 1 до n , одну из строк S или T , а также любую маленькую латинскую букву c , и заменяет i -й символ в выбранной строке на символ c .

Боб же в свой ход выбирает одну из строк, S или T , и переворачивает её, то есть делает замену S на $\text{rev}(S)$ или же T на $\text{rev}(T)$, где $\text{rev}(S)$ означает перевёрнутую строку S (т.е. $\text{rev}(S) = S_n S_{n-1} \dots S_1$).

Игра продолжается, пока строки S и T не равны. Как только строки становятся равными, игра **моментально заканчивается**.

Определим длительность игры, как суммарное количество ходов сделанное обоими игроками в процессе игры. Например, если всего Алиса сделала 2 хода, а Боб — 1 ход, то длительность такой игры равна 3.

Цель Алисы — минимизировать длительность игры, а цель Боба — максимизировать длительность игры.

Чему будет равна длительность игры, при оптимальной игре обоих игроков? Можно показать, что игра завершится за конечное число ходов.

Формат входных данных

Первая строка входных данных содержит единственное целое число n ($1 \leq n \leq 100\,000$) — длину строк S и T .

Вторая строка входных данных содержит строку S длины n , состоящую из маленьких латинских букв.

Третья строка входных данных содержит строку T длины n , состоящую из маленьких латинских букв.

Формат выходных данных

Выведите единственное число — длительность описанной игры при оптимальной игре обоих игроков.

Примеры

стандартный ввод	стандартный вывод
5 abcde abxde	1
5 hello olleo	2
2 ab cd	3
7 aaaaaaa abbbbba	9
1 q q	0

Замечание

В первом примере Алиса в свой ход может заменить третий символ строки S на x . После чего обе строки станут равны $abxde$ и игра завершится после одного хода. Так как цель Алисы минимизировать длительность игры, этот ход будет одним из оптимальных первых ходов для неё, и итоговый ответ равен 1.

Во втором примере Алиса в свой ход может заменить пятый символ строки T на h . После этого хода $S = \text{hello}$, $T = \text{olleh}$. Далее ходит Боб. В свой ход он обязан перевернуть одну из строк. Если Боб выберет строку S , то после его хода обе строки будут равны $olleh$, а если он выберет строку T , то после его хода обе строки будут равны hello . Таким образом, после представленного первого хода Алисы игра гарантированно завершится через 2 хода. Несложно показать, что стратегии гарантирующей завершение игры менее чем за 2 хода у Алисы не существует. Итого ответ равен 2.

В третьем примере Алиса в свой первый ход может заменить второй символ строки S на c . После этого хода $S = \text{ac}$, $T = \text{cd}$. Далее ходит Боб. Если Боб перевернёт строку S , то после его хода $S = \text{ca}$, $T = \text{cd}$. Тогда несложно видеть, что в этом случае Алиса может гарантированно закончить игру на 3-м ходу, заменив второй символ строки T на a , после чего обе строки станут равны ca . Если же Боб перевернёт строку T , то после его хода $S = \text{ac}$, $T = \text{dc}$. В этом случае Алиса также может гарантированно закончить игру на 3-м ходу, заменив первый символ строки S на d , после чего обе строки станут равны dc . Таким образом Алиса может гарантированно закончить игру за 3 хода вне зависимости от ходов Боба. Можно показать, что меньше чем за 3 хода, при оптимальной игре Боба, игра завершиться не может.

В пятом примере строки S и T равны, а значит игра завершится не начавшись, за 0 ходов.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты проверки ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $n \leq 2$, наберут не менее 10 баллов.

Решения, корректно работающие при $n \leq 7$, наберут не менее 40 баллов.

Решения, корректно работающие при $n \leq 2000$, наберут не менее 70 баллов.

Дополнительно, решения, корректно работающие, когда строка S состоит только из символов a , наберут не менее 10 баллов.

Задача С. Опрос на уроке

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

На урок истории к Зинаиде Викторовне пришли n учеников. На дом было задано m тем, но у учеников было мало времени для подготовки, поэтому i -й ученик выучил только темы с l_i по r_i включительно.

В начале урока каждый ученик держит свою руку на уровне 0. Учительница хочет спросить какие-то темы. Происходит это так:

- Учительница спрашивает тему k .
- Если ученик выучил тему k , то он поднимает руку на 1, а иначе опускает на 1.

Каждую тему Зинаида Викторовна может спросить **не более** 1 раза.

Найдите, какая максимальная разность между самой высокой и самой низкой рукой может быть после опроса.

Обратите внимание, рука ученика может опускаться **ниже** 0.

Формат входных данных

В первой строке содержатся два целых числа n и m ($2 \leq n \leq 200\,000, 1 \leq m \leq 10^9$) — количество учеников и количество тем соответственно.

В каждой из следующих n строк содержатся по два целых числа l_i и r_i ($1 \leq l_i \leq r_i \leq m$), обозначающих границы отрезка тем, которые выучил i -й ученик.

Формат выходных данных

Выведите одно число — максимальную разность между самой высокой и самой низкой рукой, которая может быть в классе после опроса.

Примеры

стандартный ввод	стандартный вывод
4 8 2 6 4 8 2 7 1 5	6
3 3 1 3 2 3 2 2	4

Замечание

В первом примере Зинаида Викторовна может спросить темы 6, 7, 8. Тогда рука 2-го ученика будет на высоте 3, а 4-го — на высоте -3 , то есть разность будет равна 6.

Во втором примере можно спросить про темы 1 и 3.

Система оценки

В данной задаче 20 тестов, помимо тестов из условия, каждый из них оценивается в 5 баллов. Результаты проверки ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие в случае $n \leq 10$, наберут не менее 10 баллов.

Решения, корректно работающие в случае $n \leq 100, m \leq 100$, наберут не менее 30 баллов.

Решения, корректно работающие в случае $n \leq 1000$, наберут не менее 50 баллов.

Задача D. Разрезание торта

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

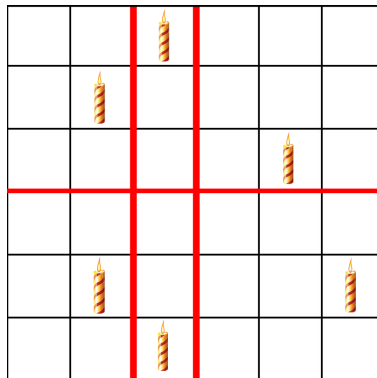
Недавно Максиму исполнилось k лет. В честь этого ему подарили торт, представляющий собой таблицу из n строк и m столбцов. На торте находятся k свечей, i -я свеча находится на пересечении x_i -й строки и y_i -го столбца (две свечи не могут стоять в одном и том же месте).

Максим хочет разрезать торт так, чтобы на каждом куске была **ровно** одна свеча. Для этого он может сделать несколько (возможно 0) горизонтальных разрезов и несколько (возможно 0) вертикальных разрезов. Каждый разрез должен проходить вдоль всего торта и должен проходить только по границам клеток таблицы.

Более формально, Максим может сколько угодно раз сделать одно из следующих действий:

- Выбрать число x такое, что $1 \leq x \leq n - 1$ и провести горизонтальный разрез, если он не был до этого проведён, отделив строки с номерами $1, \dots, x$ от строк с номерами $x + 1, \dots, n$.
- Выбрать число y такое, что $1 \leq y \leq m - 1$ и провести вертикальный разрез, если он не был до этого проведён, отделив столбцы с номерами $1, \dots, y$ от столбцов с номерами $y + 1, \dots, m$.

Определите, получится ли разрезать торт так, чтобы на каждой части была **ровно** одна свеча.



Красным цветом выделены разрезы.

Формат входных данных

Первая строка входных данных содержит три целых числа n , m и k ($1 \leq n, m \leq 10^9$, $1 \leq k \leq 10^5$) — размеры торта и количество свечей соответственно.

В i -й из следующих k строк записаны два целых числа x_i и y_i ($1 \leq x_i \leq n$, $1 \leq y_i \leq m$) — номер строки и столбца, где находится i -я свеча, соответственно.

Формат выходных данных

Выведите «No» (без кавычек), если Максим не сможет разрезать торт так, чтобы на каждой части было по одной свече.

В противном случае выведите «Yes». Во второй строке выведите два целых числа a и b — количество горизонтальных и вертикальных разрезов, которые надо сделать Максиму, соответственно.

В третьей строке выведите a **различных** целых чисел — горизонтальные разрезы, которые надо сделать Максиму, в любом порядке.

В четвёртой строке выведите b **различных** целых чисел — вертикальные разрезы, которые надо сделать Максиму, в любом порядке.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную). Например, строки «yEs», «yes», «Yes» и «YES» будут приняты как положительный ответ.

Примеры

стандартный ввод	стандартный вывод
1 1 1 1 1	Yes 0 0
2 2 4 1 1 1 2 2 1 2 2	Yes 1 1 1 1
2 2 3 1 1 1 2 2 2	No
6 6 6 2 2 1 3 3 5 5 6 6 3 5 2	Yes 1 2 3 2 3
1000000000 1000000000 4 2 1 1 2 2 3 3 2	No

Замечание

В первом примере проводить разрезы не надо.

Во втором примере, на пересечении каждой строки и каждого столбца стоит свеча, поэтому надо провести все возможные разрезы.

В четвёртом примере можно провести разрезы так, как показано на рисунке из условия.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты проверки ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие в случае $k \leq 200$, наберут не менее 30 баллов.

Решения, корректно работающие в случае $k \leq 2000$, наберут не менее 40 баллов.

Решения, корректно работающие в случае $\min(n, m) \leq 20$, наберут не менее 30 баллов.

Задача Е. Печатная машинка

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Недавно Поликарпу подарили необычную печатную машинку!

Машинка состоит из n ячеек, пронумерованных слева направо от 1 до n , и движущейся над ними каретки. В ячейках печатной машинки лежат n **различных** чисел от 1 до n , причем в каждой ячейке i изначально лежит число p_i . До всех действий каретка находится на ячейке с номером 1 и в ее буферном хранилище ничего нет. Назовем ячейку **текущей**, если каретка стоит на этой ячейке.

Она может выполнять пять типов операций:

1. Взять число из текущей ячейки и положить его в буфер каретки, если текущая ячейка не пуста, а буфер пуст.
2. Положить число из буфера каретки в текущую ячейку, если она пустая.
3. Поменять местами число, которое находится в буфере каретки, с числом, которое находится в текущей ячейке, если и буфер, и ячейка содержат числа.
4. Сдвинуть каретку с текущей ячейки i на ячейку $i + 1$ (если $i < n$), при этом число в буфере сохраняется.
5. Сбросить каретку, то есть переместить ее на ячейку с номером 1, при этом число в буфере сохраняется.

Поликарпа очень заинтересовала эта печатная машинка, поэтому он просит вас помочь разобраться в ней и задаст вам q запросов трех типов:

1. Выполнить циклический сдвиг последовательности p на k_j влево.
2. Выполнить циклический сдвиг последовательности p на k_j вправо.
3. Развернуть последовательность p .

До всех запросов, а также после каждого запроса Поликарп хочет узнать, сколько необходимо сбросов каретки для текущей последовательности, чтобы распределить числа по своим ячейкам (чтобы число i оказалось в клетке с номером i).

Обратите внимание, что Поликарп хочет только узнать минимальное количество сбросов каретки для расположения чисел по своим местам, но он не распределяет их на самом деле.

Помогите Поликарпу узнать ответы на интересующие его запросы!

Формат входных данных

Первая строка содержит одно целое число n ($1 \leq n \leq 200\,000$) — количество ячеек.

Вторая строка содержит n целых **различных** чисел p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$) — начальное расположение чисел по ячейкам.

Третья строка содержит единственное число q ($0 \leq q \leq 200\,000$) — количество запросов.

В каждой из следующих q строках содержится запросы Поликарпа:

В строке j сначала идет число t_j — тип запроса ($1 \leq t_j \leq 3$).

Если запрос типа $t_j = 1$ или $t_j = 2$, то далее, в той же строке, содержится число k_j — длина сдвига ($1 \leq k_j \leq n$).

Формат выходных данных

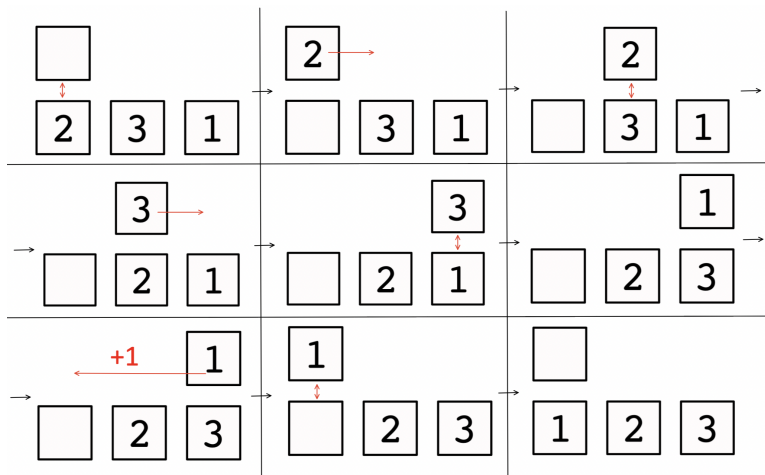
Выведите $q + 1$ число — минимальное количество сбросов каретки до всех запросов, а также после каждого запроса Поликарпа.

Примеры

стандартный ввод	стандартный вывод
3 2 3 1 0	1
3 1 2 3 2 2 1 3	0 2 1
5 3 1 2 5 4 5 1 3 3 2 3 1 4 3	3 2 1 2 1 2

Замечание

В первом примере ответ 1. По рисунку ниже можно понять, как работает каретка в этом примере.



Во втором примере последовательности, для которых нужно посчитать ответ, выглядят так:

1. До всех запросов: 1 2 3 — ответ 0.
2. После сдвига на 1 вправо: 3 1 2 — ответ 2.
3. После разворота последовательности: 2 1 3 — ответ 1.

В третьем примере последовательности до всех запросов и после каждого запроса выглядят так:

1. 3 1 2 5 4 — ответ 3.
2. 5 4 3 1 2 — ответ 2.

3. 2 1 3 4 5 — ответ 1.
4. 3 4 5 2 1 — ответ 2.
5. 1 3 4 5 2 — ответ 1.
6. 2 5 4 3 1 — ответ 2.

Система оценки

Тесты к этой задаче состоят из 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		n	m		
0	0	–	–	–	Тесты из условия.
1	16	$n \leq 6$	$q = 0$	–	
2	20	$n \leq 2000$	$q \leq 2000$	0, 1	
3	17	–	–	–	Все $p_i = i$.
4	24	–	–	1	Нет запросов типа $t_j = 3$.
5	23	–	–	0–4	

Задача F. Дерево на Манхэттене

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Участникам, использующим язык Python3, рекомендуется отправлять решения на проверку с использованием интерпретатора PyPy3.

Однажды мальчик Вова нашел корневое дерево[†] из n вершин, i -я вершина которого была соединена со своим предком p_i ($p_i < i$) ребром, с записанным на нём числом c_i . Корнем является вершина 1. Увидев его, Вова сразу же придумал интереснейшую задачу, которую вам предстоит решить.

Выкладыванием дерева на прямую назовём сопоставление каждой вершине v целого числа x_v ($1 \leq x_v \leq n$), при котором разным вершинам соответствуют разные числа, а каждое поддереву является последовательным подотрезком.

Проще говоря, выкладывание — это такая перестановка[‡] на вершинах, что для любой вершины v , если отсортировать значения x_u всех вершин u в поддереве вершины v , то образуется отрезок целых чисел $[l, r]$ для каких-то $1 \leq l \leq r \leq n$.

Стоимостью выкладывания назовём величину $\sum_{v=2}^n |x_v - x_{p_v}| \cdot c_v$.

Посчитайте минимальную стоимость выкладывания для данного вам дерева.

[†] *Дерево* — это связный неориентированный граф без циклов. *Корневое дерево* — дерево с выделенной вершиной, которую называют корнем.

[‡] *Перестановкой* длины n является массив, состоящий из n различных целых чисел от 1 до n в произвольном порядке. Например, $[2, 3, 1, 5, 4]$ — перестановка, но $[1, 2, 2]$ не перестановка (2 встречается в массиве дважды) и $[1, 3, 4]$ тоже не перестановка ($n = 3$, но в массиве встречается 4).

Формат входных данных

Первая строка содержит одно целое число n ($2 \leq n \leq 5000$) — количество вершин в дереве.

Затем следуют $n - 1$ строк, i -я из которых содержит два целых числа p_{i+1} и c_{i+1} ($1 \leq p_{i+1} \leq i, 0 \leq c_{i+1} \leq 10^{11}$) — предок $(i + 1)$ -й вершины и число, записанное на ребре из $i + 1$ в p_{i+1} .

Формат выходных данных

В единственной строке выведите одно число — минимальную стоимость выкладывания данного вам дерева.

Примеры

стандартный ввод	стандартный вывод
5 1 6 1 5 2 4 2 3	21
8 1 6 1 9 2 2 2 9 5 4 3 9 6 11	56

Замечание

Можно показать, что добиться на первом тесте из условия стоимости выкладывания меньше, чем 21 нельзя. Данная стоимость получается с помощью выкладывания $x = \{4, 3, 5, 2, 1\}$.

Аналогично, для второго теста стоимости выкладывания меньше, чем 56 получить нельзя. Пример выкладывания с данной стоимостью $x = \{6, 5, 7, 1, 4, 3, 8, 2\}$.

Система оценки

Тесты к этой задаче состоят из 6 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. Обозначим через d_i — количество сыновей вершины i , то есть количество вершин v таких что $p_v = i$.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		n	c_i	d_i		
0	0	–	–	–	–	Тесты из условия.
1	8	$n \leq 8$	–	–	0	
2	13	$n \leq 50$	–	$d_i \leq 8$	0 – 1	
3	17	$n \leq 50$	–	$d_i \leq 20$	0 – 2	
4	23	$n \leq 50$	–	–	0 – 3	
5	19	–	$c_i = 1$	–	–	
6	20	–	–	–	0 – 5	