

Задача А. Аквапарк

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Ваш друг недавно построил большой аквапарк, и собирается открыть его для общественного доступа. Доступ в аквапарк будет проходить по сеансам длительностью a минут, то есть в начале сеанса аквапарк пускает всех желающих его посетить, и через a минут сеанс заканчивается и люди покидают аквапарк.

К сожалению, бассейны аквапарка могут загрязняться, поэтому если после окончания очередного сеанса прошло не менее t минут с момента **начала** предыдущей очистки бассейнов или первого запуска аквапарка, то бассейны надо очистить. Очистка бассейнов длится b минут, а после этого сеансы посещения аквапарка продолжают. При этом изначально бассейны чистые, и в начале аквапарк готов принимать сеансы без предварительной очистки. Во время сеанса очистка бассейнов также не производится.

Вашему другу стало интересно, какое максимальное количество сеансов он сможет провести за n минут, если будет очищать бассейны сразу после конца сеанса, во время которого произошло их загрязнение.

Формат входных данных

В первой строке вводится целое число n ($1 \leq n \leq 10^{18}$) — общее время работы аквапарка.

Во второй строке вводится целое число t ($1 \leq t \leq 10^{18}$) — время от предыдущей очистки, через которое нельзя запускать новые сеансы, не очистив бассейнов.

В третьей строке вводится целое число a ($1 \leq a \leq t$) — продолжительность сеанса аквапарка.

В четвёртой строке вводится целое число b ($0 \leq b < t$) — продолжительность очистки бассейнов.

Формат выходных данных

В единственной строке выведите максимальное число сеансов, которые могли бы быть организованы за n минут работы аквапарка.

Примеры

стандартный ввод	стандартный вывод
10 5 2 3	3
10 3 1 2	5
8 7 3 5	2

Замечание

В первом примере из условия первый сеанс идёт первые 2 минуты. Далее, так как с запуска прошло 2 минуты, то второй сеанс идёт на 3 и 4 минутах работы аквапарка. Таким образом после конца второго сеанса пройдёт 4 минуты с момента запуска аквапарка, поэтому на 5 и 6 минутах работы аквапарка пройдёт третий сеанс. После окончания третьего сеанса с начала работы пройдёт 6 минут, то есть с момента запуска аквапарка прошло хотя-бы $t = 5$ минут, и значит надо очистить бассейны. Таким образом очистка будет происходить на 7, 8 и 9 минутах работы аквапарка. После этого останется лишь одна минута, за которую невозможно провести сеанс.

Во втором примере из условия первые 3 сеанса будут проходить на 1, 2 и 3 минутах. Потом пройдёт 3 минуты с момента запуска аквапарка, поэтому на минутах 4 и 5 будет очистка бассейнов, а на минуте 6 будет четвёртый сеанс. После этого пройдёт 3 минуты с начала предыдущей очистки бассейнов и значит на минутах 7 и 8 будет очередная очистка бассейнов. Далее на минуте 9 будет пятый сеанс, после него бассейны очистятся и подготовятся к следующему сеансу не успеют.

Система оценки

В данной задаче 25 тестов, помимо тестов из условия, каждый из них оценивается в 4 балла. Результаты работы ваших решений на всех тестах будут доступны сразу во время соревнования.

Решения, корректно работающие при $n, m, a, b \leq 10$, наберут не менее 20 баллов.

Решения, корректно работающие при $n, m, a, b \leq 10^6$, наберут не менее 40 баллов.

Решения, корректно работающие при $b = 0$, наберут не менее 20 баллов.

Задача В. Максимальная ширина

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

У вашего одноклассника, которого вы не очень любите за его занудство, но уважаете за его ум, были обнаружены две строки: строка t длины m и строка s длины n . Последовательность индексов p_1, p_2, \dots, p_m , где $1 \leq p_1 < p_2 < \dots < p_m \leq n$, называется *хорошей*, если $s_{p_i} = t_i$ для всех i от 1 до m . *Шириной* последовательности называется величина $\max_{i=1}^{m-1} (p_{i+1} - p_i)$, то есть максимальная разность между соседними элементами последовательности p .

Помогите однокласснику найти хорошую последовательность индексов с максимальной шириной. Одноклассник обещал вам, что строки s и t таковы, что хотя бы одна хорошая последовательность точно существует.

Формат входных данных

Первая строка входных данных содержит два числа n и m ($2 \leq m \leq n \leq 200\,000$) — длины строк s и t соответственно.

Во второй строке входных данных задана строка s , состоящая из строчных букв английского алфавита, а в третьей строке задана строка t , состоящая из строчных букв английского алфавита.

Гарантируется, что существует хотя бы одна хорошая последовательность индексов.

Формат выходных данных

Выведите одно число — максимальную ширину хорошей последовательности.

Примеры

стандартный ввод	стандартный вывод
5 3 abbbc abc	3
5 2 aaaaa aa	4
5 5 abcdf abcdf	1
2 2 ab ab	1

Замечание

В первом примере из условия существуют две хорошие последовательности с шириной 3: это $\{1, 2, 5\}$ и $\{1, 4, 5\}$.

Во втором примере из условия хорошая последовательность максимальной ширины — это $\{1, 5\}$.

В третьем примере из условия есть лишь одна хорошая последовательность — это $\{1, 2, 3, 4, 5\}$.

В четвёртом примере из условия есть лишь одна хорошая последовательность — это $\{1, 2\}$.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на первых 40 тестах будут доступны во время соревнования. Результаты работы на остальных 10 будут доступны после окончания соревнования.

Решения, корректно работающие при $n, m \leq 5$, будут набирать не менее 20 баллов.

Решения, корректно работающие при $n, m \leq 100$, будут набирать не менее 40 баллов.

Решения, корректно работающие при $n, m \leq 1000$, будут набирать не менее 60 баллов.

Решения, корректно работающие когда строка t состоит только из символов «а», будут набирать не менее 20 баллов.

Задача С. Ход гения

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Стёпа — простой парень из мегаполиса, который только начал заниматься программированием. Прочитав рекомендации на никому неизвестном сайте в далёком уголке интернета, он решил начать изучение тем с прочтения разборов недавно прошедших олимпиад. Простые задачи показались ему совсем не интересными, его внимание привлекла неизвестная ему ранее структура под названием «дерево отрезков», и он решил плотно заняться её изучением.

Через пару дней он встретился со своим более опытным другом Вацлавом, который решил проверить его знания и дал ему следующую задачу.

Даны три числа a , b , k . Найти два числа x и y ($y \leq x$), состоящие из a нулей и b единиц в двоичной системе счисления каждое, причём такие, что $x - y$ в двоичной системе содержит в себе ровно k единиц. В x и y ведущие нули запрещены.

Напомним, что двоичная система счисления устроена аналогично десятичной. А именно, в данном случае число 2 — это основание системы счисления, любое неотрицательное целое число может быть записано цифрами со значениями 0 или 1. В такой записи, если у числа длина k , и число имеет запись вида $\overline{a_{k-1}a_{k-2}a_{k-3}\dots a_1a_0}$, то величина числа равна $a_0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 + \dots + a_{k-1} \cdot 2^{k-1}$. При этом, если у числа длина больше 1, то старшая цифра не должна равняться 0, то есть, если $k > 1$, то $a_{k-1} > 0$. Но если длина равна 1, то число может состоять из единственной цифры 0.

Например, число 20 можно записать в двоичной системе счисления как $20 = 10100_2$.

Стёпа все еще не верит, что такую задачу можно решить без использования продвинутых структур данных. Помогите ему решить её во что бы то ни стало!

Формат входных данных

В единственной строке заданы три целых числа a , b , k ($1 \leq a + b \leq 200\,000$, $0 \leq a$, $1 \leq b$, $0 \leq k \leq a + b$).

Формат выходных данных

В первой строке выведите «Yes», если можно найти два подходящих числа или «No» в противном случае.

В случае, если ответ существует, во второй строке выведите число x в двоичной системе счисления, а в третьей строке выведите число y также в двоичной системе счисления.

Если возможных ответов несколько, то выведите любой.

Примеры

стандартный ввод	стандартный вывод
4 2 3	Yes 101000 100001
3 2 1	Yes 10100 10010
3 2 5	No

Замечание

В первом примере из условия $x = 101000_2 = 2^5 + 2^3 = 40_{10}$, $y = 100001_2 = 2^5 + 2^0 = 33_{10}$, $40_{10} - 33_{10} = 7_{10} = 2^2 + 2^1 + 2^0 = 111_2$. Отсюда видно, что в $x - y$ содержатся ровно 3 единицы.

Во втором примере из условия $x = 10100_2 = 2^4 + 2^2 = 20_{10}$, $y = 10010_2 = 2^4 + 2^1 = 18$, $x - y = 20 - 18 = 2_{10} = 10_2$. Ровно одна единица.

В третьем примере из условия можно показать, что ответа нет.

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 $a + b \leq 10$, наберут не менее 20 баллов.

Решения, корректно работающие при $a + b \leq 200$, наберут не менее 40 баллов.

Решения, корректно работающие при $a + b \leq 3000$, наберут не менее 50 баллов.

Решения, корректно работающие при $b \leq 20$, наберут не менее 40 баллов (включая 20 баллов за тесты, в которых выполнено $a + b \leq 10$).

Решения, корректно работающие при $k \leq 20$, наберут не менее 30 баллов (включая 20 баллов за тесты, в которых выполнено $a + b \leq 10$).

Решения, корректно работающие только на тестах из условия и тестах с ответом «No», оцениваются в 0 баллов.

Задача D. Почти отказоустойчивая база данных

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Вы храните в базе данных массив длины m . Чтобы обезопасить данные от случайного повреждения, база данных хранит не одну, а n физических копий хранимой информации.

К сожалению, недавно в базе данных произошла масштабная авария, которая потенциально изменила информацию в каждой копии.

Предполагается, что инцидент поменял не более одного элемента в каждой копии. Вам нужно восстановить исходный массив из текущего состояния базы данных.

Если существует несколько способ восстановления, найдите любой. Если нет ни одного массива, отличающегося от каждой копии не более чем в одной позиции, сообщите об этом.

Формат входных данных

В первой строке заданы два целых числа n и m ($2 \leq n$, $1 \leq m$, $n \cdot m \leq 250\,000$) — количество физических копий исходной информации и размер массива исходной информации, соответственно.

В каждой из следующих n строк задан массив из m чисел s_j ($1 \leq s_j \leq 10^9$) — очередная физическая копия в базе данных после аварии.

Формат выходных данных

Если можно восстановить массив, который мог бы представлять исходную информацию, то выведите в первой строке слово «Yes», и во второй строке сам массив из ровно m чисел от 1 до 10^9 .

Если возможных ответов несколько, то выведите любой.

Если ни одного ответа нет, то выведите в единственной строке слово «No».

Примеры

стандартный ввод	стандартный вывод
3 3 1 100 100 1 1 100 10 10 100	Yes 1 10 100
8 7 1 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1	Yes 1 1 1 1 1 1 1
2 3 1 1 1 2 2 2	No

Замечание

В первом примере из условия массив $[1, 10, 100]$ отличается от каждого из трёх данных физических копий ровно в одной позиции.

Во втором примере из условия массив $[1, 1, 1, 1, 1, 1, 1]$ равен восьмой физической копии и отличается от каждой из первых семи физических копий ровно в одной позиции.

В третьем примере из условия невозможно найти какой-нибудь массив, который отличается не более чем в одной позиции от обеих физических копий после аварии.

Система оценки

В данной задаче 100 тестов, помимо тестов из условия, каждый из них оценивается в 1 балл. Результаты работы ваших решений на первых 70 тестах будут доступны во время соревнования. Результаты работы на остальных 30 будут доступны после окончания соревнования.

Пусть $S = \max_{1 \leq i \leq n, 1 \leq j \leq m} s_{i,j}$.

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

Решения, корректно работающие при $n, m \leq 10$, наберут не менее 20 баллов.

Решения, корректно работающие при $n, m \leq 500$, наберут не менее 40 баллов.

Решения, корректно работающие при $S \leq 50$, наберут не менее 40 баллов (включая 10 баллов за тесты, в которых выполнено $S \leq 2, n, m \leq 10$).

Задача Е. Древесные жабы (и лягушки)

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Древесные жабы живут на деревьях. Как известно, дерево — это связный граф, такой, что между любыми двумя вершинами существует ровно один простой путь. Более строго, дерево состоит из вершин и соединяющих пары вершин рёбер. Из каждой вершины переходя по рёбрам можно добраться до любой другой вершины, при этом, если не переходить по одному ребру несколько раз, то путь из одной вершины в другую будет единственным. Можно показать, что в таких графах ровно $n - 1$ ребро, где n — число вершин в графе.

Так вот, задача вообще-то про жаб. В каждой из вершин сидит по жабе. Первая жаба сидит в вершине с номером 1, вторая жаба — в вершине с номером 2 и так далее. За одну единицу времени каждая из жаб может перейти по ребру в соседнюю вершину.

Жабы часто завидует другим жабам из-за того, что у них мало букашек.

Жабы успокаивают себя мыслью о том, что хоть у них букашек меньше, но они всё равно достаточно крутые жабы. Крутостью жабы назовём сумму номеров всех вершин, до каждой из которых в отдельности эта жаба может добраться строго быстрее первой жабы, если они начнут свой путь в одно и то же время.

Вас, как эксперта по жабам (и лягушкам), просят ответить на m запросов о крутости той или иной жабы, а именно, на каждый из m запросов, который задаётся одним числом u , вам необходимо сообщить крутость u -й жабы.

Формат входных данных

В первой строке входного файла через пробел вводятся два числа n ($1 \leq n \leq 100\,000$) — количество вершин в дереве и m ($1 \leq m \leq 100\,000$) — количество запросов.

В следующих $n - 1$ строках описаны рёбра. Ребро задаётся числами x_i и y_i — номерами вершин, которые оно соединяет ($1 \leq x_i, y_i \leq n, x_i \neq y_i$). Гарантируется, что между любыми двумя вершинами существует единственный путь.

В следующих m строках описаны запросы. В i -й из них задано число u_i — номер очередной жабы, для которой надо определить её крутость. Вершины (и жабы) нумеруются с единицы.

Формат выходных данных

На каждый из запросов u_i выведите по одному числу в отдельной строке — крутость жабы, сидящей в вершине u_i .

Пример

стандартный ввод	стандартный вывод
7 5	27
1 2	25
2 3	18
3 4	25
3 5	0
5 6	
5 7	
2	
4	
7	
3	
1	

Замечание

В примере из условия ответ на первый запрос — 27, так как жаба, сидящая в вершине 2 может быстрее жабы номер 1 добраться до вершин 2, 3, 4, 5, 6, 7, то есть 27 в сумме.

Ответ на второй запрос — 25, так как жаба, сидящая в вершине 4 может быстрее жабы номер 1 добраться до вершин 3, 5, 6, 7.

Ответ на третий запрос — 18, так как жаба, сидящая в вершине 7 может быстрее жабы номер 1 добраться до вершин 5, 6, 7 (до вершин 3 и 4 они доберутся за одно и то же время).

Ответ на последний запрос — 0, так как жаба, сидящая в вершине 1 не сможет добраться ни до одной вершины быстрее, чем другая жаба, сидящая в вершине 1.

Система оценки

В данной задаче 50 тестов, помимо тестов из условия, каждый из них оценивается в 2 балла. Результаты работы ваших решений на первых 35 тестах будут доступны во время соревнования. Результаты работы на остальных 15 будут доступны после окончания соревнования.

Решения, корректно работающие при $n, m \leq 100$, будут набирать не менее 20 баллов.

Решения, корректно работающие при $n, m \leq 1000$, будут набирать не менее 50 баллов.

Решения, корректно работающие на графах, в которых из вершины не может исходить больше 2 рёбер, будут набирать не менее 10 баллов.

Решения, корректно работающие на тестах, в которых расстояние от каждой из вершин u_i до вершины 1 не превосходит 20, будут набирать не менее 10 баллов.

Задача F. Сложная задача

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	768 мегабайт

У Егора есть большой лист клетчатой бумаги и не менее большое домашнее задание по математике. Егор, как недобросовестный ученик, планирует списать домашнюю работу с сервиса Титан Бета.

Решение задачи Егора в системе Титан Бета состоит из n формул-шагов, каждую из которых он должен перенести на лист. Разумеется, формулы в листе должны идти по порядку — слева направо. Если в какой-то момент написать следующую формулу нельзя — её обязательно надо перенести на следующую строку. Но также разрешено делать переносы и в произвольном месте решения, на усмотрение ученика, однако формула должна быть написана целиком на одной строке.

Поскольку в формулах могут возникать дробные выражения, высоты формул бывают разными. Егор относится к формулам, как к бессвязному набору символов, который влезает в какой-то клетчатый прямоугольник; а именно, i -я формула занимает прямоугольник w_i клеток в ширину и h_i клеток в высоту.

Чтобы строки читались аккуратно, Егор выравнивает все формулы, записанные в одной строке, по нижней границе строки. Таким образом, высота строки равна высоте самой высокой формуле строки. Ширина же строки равна суммарной ширине формул в строке. Суммарной высотой решения назовем суммарную высоту всех строк.

Ширина листа бумаги составляет ровно m клеток. Егора заинтересовало, какую наименьшую высоту может иметь его решение? Поскольку ему надо делать домашнюю работу, то ответ на этот вопрос предстоит найти вам.

Формат входных данных

В данной задаче ввод устроен необычным образом — для тестов последней группы часть входного файла вам придется генерировать самостоятельно.

В первой строке вводятся три числа n, m, k ($1 \leq n \leq 10^7$, $1 \leq m \leq 10^{11}$, $1 \leq k \leq \min(500\,000, n)$) — общее количество формул, ширина листа, количество формул, высота и ширина для которых будут введены далее.

В следующих k строках вводится по три целых числа i_j, w_{i_j}, h_{i_j} ($1 \leq i_j \leq n$, $1 \leq w_{i_j}, h_{i_j} \leq m$). Первое число — номер одной из формул Егора. Следующие два числа — ширина и высота этой формулы соответственно.

Гарантируется, что индексы i_j возрастают с каждой следующей строкой.

В последней строке вводятся 4 числа a, b, c, d ($1 \leq a, c \leq \min(10\,000, m)$, $0 \leq b, d \leq m$) — вспомогательные параметры, которые отвечают за генерацию остальных входных данных.

Для тех формул, номера которых не были перечислены во входных данных, высоту и ширину придётся генерировать самостоятельно. А именно, для любого i , не перечисленного во входных данных, $w_i = ((w_{i-1} \cdot a + b) \bmod m) + 1$, $h_i = ((h_{i-1} \cdot c + d) \bmod m) + 1$. Гарантируется, что $i = 1$ присутствует во входных данных (соответственно, размеры всех формул определяются однозначно).

Формат выходных данных

Выведите одно число — минимальную высоту листа шириной m , в который получится уместить все формулы.

Примеры

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

Замечание

В первом примере из условия оптимальным вариантом размещения будет тот, при котором первая формула находится на отдельной строке, а вторая и третья — на следующей. Таким образом, суммарная высота получается равной $2 + 3 = 5$ клеткам.

Во втором примере из условия прямоугольники имеют вид $[(2, 3), (1, 2), (3, 1), (2, 3)]$. Третий прямоугольник обязательно займет целую строку. Четвертая формула, таким образом, займет свою отдельную строчку, а первая и вторая формулы сгруппируются. Получившиеся строки будут вместе занимать $3 + 1 + 3 = 7$ клеток в высоту.

Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Дополнительные ограничения	Необх. группы	Комментарий
		n		
0	0	–	–	Тесты из условия.
1	12	$n \leq 10, n = k$	–	
2	13	$n \leq 100, n = k$	1	
3	14	$n \leq 3000, n = k$	1, 2	
4	7	$n \leq 500\,000, n = k$	–	$h_i < h_{i+1}$
5	8	$n \leq 500\,000, n = k$	–	$h_i > h_{i+1}$
6	22	$n \leq 500\,000, n = k$	1–5	
7	22	$n \leq 5 \cdot 10^6$	0–6	
8	2	–	0–7	Offline-проверка.