

Задача 1. Просто сложение

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

*Ноль, целковый, полушка,
четвертушка, осьмушка, пудовичок,
медячок, серебрячок, золотничок,
девятичок, десятичок.*

— Юрий Степанович Рыбников,
Счёт древних русов

Степан Юрьевич, ученик выдающегося Юрия Степановича, решил на основе счета древних русов разработать систему сложения современных русов, дабы ускорить вычисления в существующих математических моделях. Сказано — сделано. И вот перед ним на столе лежит лист бумаги, на котором написана таблица суммирования новой системы. Осталось только научить компьютер складывать числа согласно этим правилам, и...

В программу закралась ошибка, и подсчет выполняется неправильно, ставя под угрозу результаты тысяч экспериментов. Нужно срочно что-то делать. Помогите Степану Юрьевичу написать новую программу, которая не будет допускать ошибок в вычислениях.

Сложение современных русов работает следующим образом. Пусть складываются два числа. Числа записаны в десятичной системе счисления. Процесс вычисления соответствует правилам вычисления чисел «в столбик».

- Числа выравниваются по младшим разрядам.
- Если числа имеют одинаковую длину, то результирующее число тоже будет иметь такую же длину, а в каждом разряде будет записана цифра из заданной таблицы, которая определяется так: если в первом числе в k -м разряде находится цифра, равная i , а во втором числе в этом же k -м разряде записана цифра, равная j , то в k -й разряд результата будет записана цифра, которая находится на пересечении i -ой строки и j -го столбца таблицы ($0 \leq i, j \leq 9$).

- В случае, если длины чисел различные, то в число меньшей длины слева от значащих цифр вместо ведущих нулей дописывается специальная цифра '@'. Правила сложения десятичных цифр со специальной цифрой тоже задаются табличкой.
- Правило сложения двух цифр '@' не определено

Ваша задача — вывести сумму двух заданных чисел.

Формат входных данных

В первых десяти строках входного файла записано по 10 цифр через пробел — таблица, разработанная Степаном Юрьевичем.

Сначала описываются правила сложения с 0 из первого числа, далее с 1 и так далее до 9.

В следующих двух строках задано по 10 цифр — результаты сложения цифры '@' из первого числа с десятичной цифрой из второго и, соответственно, результаты сложения десятичной цифры из первого числа с цифрой '@' из второго.

В следующей строке задано целое число Q — количество заданий ($Q \leq 10^3$).

Далее в Q строках заданы по два целых неотрицательных числа, которые нужно сложить. Длина каждого числа не превышает 10^3 символов.

Формат выходных данных

В выходной файл нужно вывести Q строк, содержащих по одному числу — результату сложения соответствующей пары чисел.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	20	Числа не превышают 10^{18}	1
3	30	Числа имеют одинаковую длину	1
4	50	$Q \leq 10^3$, длины чисел варьируются от 1 до 10^3 символов	1, 2, 3

Примеры

input.txt	output.txt
0 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 0 2 3 4 5 6 7 8 9 0 1 3 4 5 6 7 8 9 0 1 2 4 5 6 7 8 9 0 1 2 3 5 6 7 8 9 0 1 2 3 4 6 7 8 9 0 1 2 3 4 5 7 8 9 0 1 2 3 4 5 6 8 9 0 1 2 3 4 5 6 7 9 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 4 1 1123456789 1 1 3 3 00 7	1123456780 2 6 07
8 2 4 1 2 7 2 8 0 6 3 9 4 9 9 5 2 2 8 0 8 4 5 9 8 7 8 4 6 5 7 5 4 1 6 9 1 9 1 2 8 6 6 5 5 7 2 1 5 6 6 8 7 2 0 0 7 3 1 0 2 7 8 3 0 7 4 6 3 8 9 3 3 6 1 9 3 8 0 7 6 3 8 9 0 2 1 0 9 5 0 3 3 0 5 7 9 4 2 2 0 5 3 9 8 0 7 4 0 6 6 9 8 7 5 5 0 8 5 6 5 693 9072 9 3962347979 54436397 4286675697 0 8347 62 118221	6244 9673984642 8312297128 0988 550384

Задача 2. Поиск «суфлёра»

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Учитель математики Семён Владимирович в конце каждой четверти устраивает своим ученикам устный экзамен. Не секрет, что существуют неблагоприятные личности, прибегающие к различным хитростям, чтобы сдать экзамены с меньшими усилиями.

Семён Владимирович подозревает, что его ученикам помогает сдавать экзамен некий «суфлер», который пытается подорвать учебный процесс, и специально подсказывает им неправильные ответы.

Чтобы понять, существует «суфлёр» или нет, учитель вычисляет подозрительность ответа каждого ученика.

Правильный ответ и ответы всех учеников — это строки одинаковой длины. Подозрительность ответа равна длине максимальной последовательности подряд идущих символов, которые отличаются в ответе ученика от соответствующей последовательности в правильном ответе.

Формально, пусть $p_1p_2\dots p_N$ — правильный ответ, $s_1s_2\dots s_N$ — ответ ученика. Тогда подозрительность ответа равна $\max_{1 \leq l \leq r \leq N} r - l + 1$ среди таких l и r , что $p_l \neq s_l, p_{l+1} \neq s_{l+1}, \dots, p_r \neq s_r$.

Если максимальная подозрительность среди всех ответов достигает порога R , экзаменатор понимает, что «суфлёр» все-таки существует. После этого экзамен останавливается, и все ученики отправляются на пересдачу. Помогите учителю понять, существует «суфлер» или нет.

Формат входных данных

В первой строке входного файла записаны три целых числа N, M, R — длина одного ответа, количество учеников, сдающих экзамен, и порог подозрительности лектора ($1 \leq N, M, N \cdot M, R \leq 3 \cdot 10^5$).

Далее следуют $M + 1$ строк, каждая из которых состоит из N строчных английских букв. В первой строке записан правильный ответ, в следующих строках — ответы учеников.

Формат выходных данных

Для каждого ответа ученика в соответствующую строку выходного файла нужно вывести его подозрительность.

В последней строке должно быть записано слово YES, если «суфлёр» существует, и NO — иначе.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	42	$1 \leq N, M \leq 100$	1
3	58	Нет дополнительных ограничений	1, 2

Пример

input.txt	output.txt
4 4 3	1
file	3
fall	2
byte	0
ball	YES
file	

Задача 3. Вадим и лимонад

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды, для Java — 4 секунды
Ограничение по памяти:	768 мегабайт

*«If life gives you lemons, make
lemonade!»*

— *«Если судьба кормит вас
сплошными лимонами, то ищите
способ сделать из них лимонад...»*

— Народная мудрость

Знаменитый блогер-программист Вадим очень любит лимонад. Он является активным сторонником здорового образа жизни, и поэтому может каждый день выпивать полностью только одну бутылку лимонада — ни больше, ни меньше. Однако, за Вадимом было замечено странное свойство: выпив V нанолитров лимонада качества q , на следующий день ему хочется выпить $V \cdot q$ нанолитров. Причем, чтобы утолить жажду в первый день, Вадиму достаточно выпить любое ненулевое количество лимонада.

Фан-клуб Вадима решил подарить ему набор бутылок с лимонадом такой, чтобы Вадим смог как можно дольше не испытывать жажды. При этом его фанаты хотят потратить как можно меньше денег.

В магазине продаются пустые бутылки без лимонада. Для каждой бутылки известны ее объем и стоимость. По счастливой случайности у одного из членов фан-клуба есть знакомый на фабрике лимонада. Поэтому за лимонад не придется платить, однако весь лимонад будет одинакового качества, и это качество может быть заказано производителям фан-клубом. Качество лимонада всегда измеряется положительным числом, но, возможно, не целым.

Так как бутылок в магазине может быть много, то вас попросили написать программу, определяющую самую дешёвую последовательность бутылок, при помощи которой Вадим сможет максимально долго утолять жажду.

Формат входных данных

В первой строке входного файла записано число N — количество бутылок в магазине ($1 \leq N \leq 5000$).

Вторая строка содержит объемы бутылок в магазине. Это N целых положительных чисел, где V_i — объем i -й бутылки ($1 \leq V_i \leq 10^{18}$, $1 \leq i \leq N$).

В третью строку записаны, соответственно, стоимости этих бутылок. Это также N целых положительных чисел, где C_i — стоимость i -й бутылки ($1 \leq C_i \leq 10^6$, $1 \leq i \leq N$).

Формат выходных данных

В первую строку выходного файла нужно вывести два числа — количество бутылок и стоимость самого дешёвого набора бутылок, при помощи которого Вадим сможет максимально долго утолять жажду.

Во вторую строку нужно вывести номера бутылок такого набора в порядке, в котором Вадим их должен пить.

Если таких наборов несколько, выведите любой.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	15	$1 \leq N \leq 20; 1 \leq V_i \leq 10^9$	1
3	35	$1 \leq N \leq 500; 1 \leq V_i \leq 10^9$	1, 2
4	40	$1 \leq V_i \leq 10^9$	1, 2, 3
5	10	Нет дополнительных ограничений	1, 2, 3, 4

Примеры

input.txt	output.txt
5 1 7 4 8 2 1 2 3 4 5	4 13 1 5 3 4
4 1 1 2 2 10 8 7 7	2 14 3 4

Замечание

Качество лимонада может быть меньше 1, тогда на следующий день Вадиму захочется выпить меньше лимонада, чем в текущий.

Задача 4. Вадим и телефонный звонок

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	3 секунды, для Java — 5 секунд
Ограничение по памяти:	256 мегабайт

— Ты опять *не* выходишь на связь,
Вадим?

Думая о том, что завтра он снова сможет полакомиться любимым лимонадом из очередной бутылки, подаренной фанатами, Вадим услышал телефонный звонок. Вадим взглянул на телефон — звонит школьный товарищ, с которым он давно не общался, а о многом так хочется поговорить... Он немедленно схватил трубку, ответил на вызов и ... в ответ услышал лишь шипение. Очевидно — плохая связь. К нему тотчас же нагрнуло воспоминание из детства, городская легенда, гласившая о том, что якобы, если произнести определенные слова во время разговора, то связь улучшится.

Вадим откопал у себя старый блокнот, в который он записывал слова, которые, как ему казалось, улучшают качество связи. Он бегло произнес пару слов из блокнота и, о чудо, связь улучшилась. После продолжительного разговора ему стало интересно, сколькими способами он может произнести слова, чтобы увеличить качество связи на некоторое значение. Внутреннее чувство тревоги подсказывает ему, что превосходить некоторый порог может быть опасно. Ведь всем известно, что лучшее — враг хорошего. Связь может вообще оборваться.

Формат входных данных

В первой строке входного файла записано целое число N — количество слов в блокноте ($1 \leq N \leq 10^5$).

В следующей строке заданы N чисел, каждое из которых означает, на сколько улучшается связь при произнесении соответствующего слова из блокнота ($1 \leq a_i \leq 99, 1 \leq i \leq N$).

В последней строке записаны два числа L и R — минимальное значение, на которое необходимо улучшить качество связи и порог, превосходить который чрезвычайно опасно ($1 \leq L \leq R \leq 10^{18}$).

Формат выходных данных

В выходной файл необходимо вывести одно число — количество различных способов произнести слова из блокнота, чтобы улучшить связь, не превосходя заданный порог.

Поскольку ответ может быть достаточно большим, выведите остаток от деления его на число 998244353.

Два способа считаются различными, если в некоторый момент времени были произнесены различные слова.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	6	$1 \leq N \leq 5; 1 \leq a_i \leq 10; 1 \leq L \leq R \leq 10$	1
3	14	$1 \leq N \leq 100; 1 \leq a_i \leq 99; 1 \leq L \leq R \leq 100$	1, 2
4	12	$1 \leq N \leq 10^5; 1 \leq a_i \leq 99; 1 \leq L \leq R \leq 10^6$ среди a_i не более 2 различных значений	1
5	18	$1 \leq N \leq 10^5; 1 \leq a_i \leq 99; 1 \leq L \leq R \leq 5 \cdot 10^4$	1, 2, 3
6	20	$R - L \leq 100$	1, 2, 3
7	30	Нет дополнительных ограничений	1, 2, 3, 4, 5, 6

Примеры

input.txt	output.txt
2 1 2 1 3	6
1 2 3 3	0
2 1 2 1 10	231

Замечание

Пусть в первом примере есть два слова: A и B соответственно, тогда допустимыми будут следующие способы: A, B, AA, AB, BA, AAA .

Во втором примере не существует ни одного способа, удовлетворяющего условию.

Задача 5. Вадим и город E

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда, для Java — 2 секунды
Ограничение по памяти:	512 мегабайт

После того, как Вадим переехал из города N в город E, за ним стали охотиться фанаты. Вадим — человек занятой, поэтому не хочет тратить свое время на диалоги. Но если так случится и фанат его встретит, то Вадим не сможет отказаться от диалога с ним.

E — город из нескольких площадей, соединенных между собой улицами таким образом, что для каждой пары площадей, соединенных улицей, существует не более одного **другого** пути, связывающего эти площади. При этом между любыми двумя площадями существует путь. **Особенность проектирования города E заключается в том, что начав свой путь с любой площади и обойдя три другие различные, невозможно вернуться на начальную площадь по другой улице напрямую.**

В самом начале Вадим находится на площади с номером 1, а фанаты на площадях $a_1 \dots a_k$. Перебежать с одной площади на другую возможно, только если они соединены улицей. За один шаг Вадим перебегает на соседнюю площадь, только после этого на свои соседние площади перебегают фанаты, после чего снова перебегает Вадим, после него фанаты, и так далее. Ни Вадим, ни фанаты не могут остаться на текущей площади. Если хотя бы один из фанатов окажется с Вадимом на одной площади, то Вадиму придется общаться с ними.

Может ли Вадим бегать от своих преследователей таким образом, чтобы они его никогда не поймали?

Формат входных данных

В первой строке входного файла записаны три целых числа N , M , K — количество площадей, количество улиц и количество фанатов, ищущих Вадима ($2 \leq N \leq 2 \cdot 10^5$, $N - 1 \leq M \leq 10^6$, $0 \leq K \leq 100$).

Во второй строке заданы K чисел a_i — номера площадей, на которых находятся фанаты изначально ($2 \leq a_i \leq N$, $1 \leq i \leq K$). Строка пустая, если $K = 0$.

В последних M строках записано по два числа — номера площадей u , v , связанных дорогой ($1 \leq u, v \leq N$; $u \neq v$).

Между любыми двумя площадями существует не более одной улицы.

Формат выходных данных

В выходной файл необходимо вывесит слово YES, если Вадим может бесконечно бегать по городу E от фанатов, иначе нужно вывести слово NO.

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	20	$0 \leq K \leq 2$; $2 \leq N \leq 100$	1
3	30	$0 \leq K \leq 2$; $2 \leq N \leq 1000$	1, 2
4	50	Нет дополнительных ограничений	1, 2, 3

Примеры

input.txt	output.txt
5 5 1 2 1 2 2 3 3 4 4 5 5 1	YES
5 5 2 4 5 1 2 2 3 3 4 4 5 5 1	NO

Задача 6. ПАДЪЁМ

Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Программирование — вторая грамотность

академик Андрей Петрович Ершов

Вадима, как известного блогера-программиста, пригласили прочитать лекцию школьникам о том, как важно уметь программировать. Лекция назначена на первый урок, в зале собрали всех учеников старших классов. К сожалению, не все школьники знают, кто такой Вадим, и не все еще полюбили программирование. Они решили в это время поспать. Вадим хочет, чтобы его выступление послушало как можно больше учеников. Нужно разбудить школьников. Для этого он решил крикнуть слово «ПАДЪЁМ» достаточно громко, чтобы школьники его услышали.

Зал можно представить на плане в виде прямоугольника размером W на H , а школьников и Вадима — в виде точек внутри него. Школьник может находиться в двух состояниях: спит или слушает. Спящий школьник просыпается, если слышит звук громкостью не менее P . Слушающий ученик, услышав крик Вадима громкости не менее P , начинает смеяться с громкостью L , чем также может разбудить соседних учеников. При этом школьники достаточно умны, поэтому ученик может засмеяться только услышав крик Вадима, но не смех другого школьника. Громкость звука линейно убывает на K единиц за каждую единицу расстояния. Если на школьника действует несколько звуков, то их громкость **не** складывается.

Помогите Вадиму крикнуть с наименьшей громкостью, чтобы при этом разбудить всю аудиторию и привлечь к себе внимание.

Формат входных данных

В первой строке входного файла заданы числа W , H , N и K , P и L — размеры зала, количество школьников, коэффициент, с которым уменьшается громкость, минимальная громкость для пробуждения и громкость смеха ($1 \leq W, H \leq 10^6$, $1 \leq N \leq 2 \cdot 10^5$, $1 \leq K \leq 5$, $0 \leq P, L \leq 10^7$).

Во второй строке заданы x_0, y_0 — координаты Вадима ($0 \leq x_0 \leq W$, $0 \leq y_0 \leq H$).

В следующих N строках записано по три числа: x_i, y_i, s_i — координаты очередного школьника и его состояние ($0 \leq x_i \leq W$, $0 \leq y_i \leq H$, $s_i \in \{0, 1\}$, $s_i = 0$ означает, что школьник спит, $s_i = 1$ — слушает). Все координаты целочисленные.

Гарантируется, что никакие два ученика, а также Вадим и какой-то школьник не могут находиться в одной точке.

Формат выходных данных

В выходной файл необходимо вывести одно число — минимальную требуемую громкость. Ответ будет считаться верным, если его относительная или абсолютная погрешность по сравнению с правильным не превышает 10^{-9} .

Система оценки

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой под-

задачи и необходимых подзадач успешно пройдены. Баллы в последней группе начисляются за каждый пройденный тест.

Подзадача	Баллы	Ограничения	Необходимые подзадачи
1	0	Тесты из условия	
2	20	$1 \leq N \leq 2 \cdot 10^4; 1 \leq L, P \leq 10^7$	1
3	5	$1 \leq N \leq 2 \cdot 10^5; 1 \leq L, P \leq 10^7; \text{все } s_i = 0$	1
4	20	$1 \leq N \leq 2 \cdot 10^5; 1 \leq L, P \leq 10$	1
5	55	$1 \leq N \leq 2 \cdot 10^5; 1 \leq L, P \leq 10^7$	1, 2, 3, 4

Примеры

input.txt	output.txt
8 1 3 1 1 3 6 0 5 1 0 4 1 1 3 0 0	3.23606797749978980505
1 1 1 1 1 1 0 0 1 1 1	0.00000000000000000000
1 1 1 1 1 1 0 0 1 1 0	2.41421356237309492343
4 3 6 4 1 8 2 1 3 3 0 0 0 0 3 0 0 1 0 0 2 0 1 3 1 0	9.94427190999915922021

Замечание

Первый тест представлен на рисунке.

Чёрная точка $(6, 0)$ — позиция Вадима, красные точки — позиции спящих школьников, зеленая точка $(4, 1)$ — позиция единственного бодрствующего школьника.

В данном случае Вадиму достаточно крикнуть с громкостью $1 + \sqrt{5}$, заставив смеяться бодрствующего школьника, который в свою очередь своим смехом разбудит обоих спящих товарищей.

