

Задача 1. Ход Иннокентия

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

*Иннокентий — настоящий мыслант
гигысли, способен одновременно
сыграть тысячу партий, и не
проиграть ни одну от детского мата,
ведь он всегда играет белыми.*

Иннокентий только начал учиться играть в шахматы. Чтобы натренировать понимание ходов фигур и стратегию игры его учитель предложил одно простое упражнение.

Иннокентий ходит специальной фигурой. Фигура может сходить на любую из четырех смежных по стороне клеток, либо на любую клетку вертикали **Е**, но при условии, что она стоит на этой вертикали или на смежных с ней вертикалях **Д** и **Ф**. Важно заметить, что фигура обязательно должна сходить, т.е. она не может остаться на первоначальной клетке.

Ученику хочется превзойти мастера, а для этого ему нужно быстро отвечать на вопрос — в какое количество клеток может попасть фигура за один ход. Чтобы также быстро проверять ответы Иннокентия, его учитель просит вас написать программу.

Формат входных данных

В первой строке входного файла записаны два целых числа N и M — размеры шахматной доски, где N — число горизонталей, а M — вертикалей, соответственно ($8 \leq N \leq 1000$, $8 \leq M \leq 1000$).

Во второй строке входного файла записаны два числа x, y — позиция удивительной фигуры ($1 \leq x \leq N$, $1 \leq y \leq M$).

Формат выходных данных

В выходной файл необходимо вывести одно целое число — количество клеток, в которые фигура может попасть за один ход.

Система оценки

За каждый пройденный тест начисляется 4 балла.

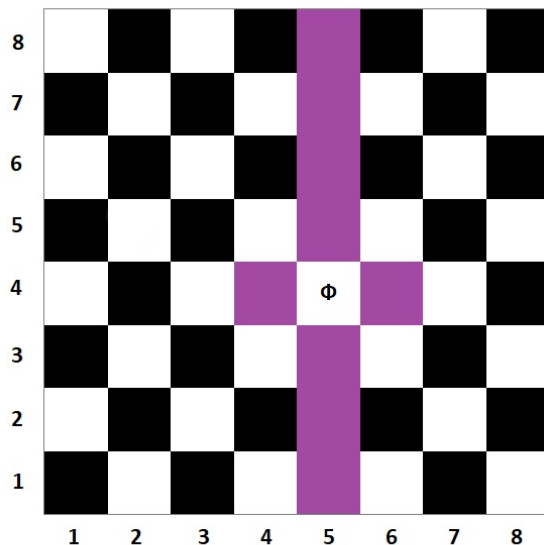
Примеры

<code>input.txt</code>	<code>output.txt</code>
8 8 1 1	2
8 8 4 5	9

Замечание

Вертикалям **Д**, **Е** и **Ф** соответствуют номера **4**, **5** и **6**.

Клетки, на которые фигура во втором тесте может попасть за один ход, изображены на рисунке:



Задача 2. Пирожки

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

— Положили тебе пирожки с
капустой, картошкой, мясом и гвозди.
— Зачем?
— Победать.
— А гвозди?
— Так вот же они!

Борис решил быстро перекусить между занятиями и зашёл в столовую. В столовой, как всегда, продавали его любимые пирожки. Но в этот раз все пирожки с разными начинками оказались на одном подносе, а отличить их между собой невозможно.

Сразу же в голову Бориса пришла задача — какое минимальное количество пирожков нужно съесть, чтобы гарантированно попробовать все их виды. Чтобы эту задачу точно можно было решить, Борис узнал у повара и кассира, сколько пирожков каждого типа лежит на подносе.

Вам он предлагает за пирожок с полки помочь ему узнать ответ на задачу.

Формат входных данных

Первая строка входного файла содержит одно целое число T — количество тестов в задаче.

Оставшиеся данные во входном файле состоят из T блоков, каждый ровно по две строки.

В первой строке блока записано число K_i — количество видов пирожков ($1 \leq i \leq T$, $1 \leq K_i \leq 10^6$).

Во второй строке блока содержится K_i чисел a_{ij} , разделенных пробелом — количество испеченных пирожков определенного типа ($1 \leq a_{ij} \leq 10^9$, $1 \leq j \leq K_i$).

Гарантируется, что сумма K_i по всем i не превосходит 10^6 .

Формат выходных данных

В выходной файл необходимо вывести T целых чисел, по одному на строке — минимальное количество пирожков которое нужно съесть, чтобы гарантированно попробовать все их виды.

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	10	$\sum_{i=1}^T K_i \leq 10^4; 1 \leq a_{ij} \leq 10^5$	1
3	15	$\sum_{i=1}^T K_i \leq 10^4; 1 \leq a_{ij} \leq 10^9$	1, 2
4	75	Нет дополнительных ограничений	1, 2, 3

Пример

input.txt	output.txt
2	5
2	4
4 4	
3	
1 2 1	

Задача 3. Просто сложение

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
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 4. 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 неблагоразумные 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 $\max_{1 \leq l < r \leq N} r - l + 1$ 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 5. Вадим и лимонад

Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
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.