Задача A. Дружелюбные ладьи

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Ладья — шахматная фигура, которая может двигаться на любое число клеток по горизонтали или по вертикали при условии, что на её пути нет фигур. Ладья бьет шахматную фигуру, если фигура находится с ладьей на одной вертикали или горизонтали. В одной клетке шахматного поля может находится не более одной ладьи.

Вам даны k ладей и шахматная доска с размером $n \times m$. Требуется расставить данные ладьи на этой доске так, чтобы они не били друг друга.

Формат входных данных

В единственной строке даны три натуральных числа n, m и k — размеры поля и число ладей, соответственно $(1 \leqslant n, m, k \leqslant 100)$.

Формат выходных данных

Если расставить k ладей на поле $n \times m$ невозможно, выведите строку `Impossible`.

Если хотя бы одна искомая расстановка существует, выведите `Possible`. А затем выведите n строк по m символов в каждой — описание итоговой расстановки ладей на поле. В i-й строке j-м символом выведите «`*`», если клетка (i, j) содержит ладью, и «`.`», если соответствующая клетка вашей расстановки пуста.

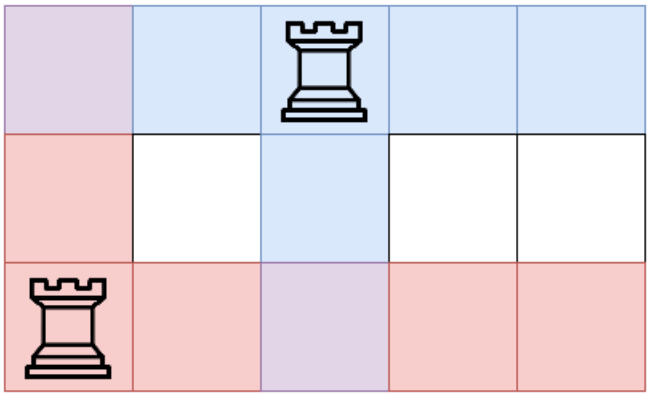
Если подходящих расстановок несколько, можно вывести любую из них.

Примеры

стандартный ввод	стандартный вывод
1 2 1	Possible *.
3 3 100	Impossible
3 5 2	Possible ..*.. ..... *....

Замечание

Иллюстрация расстановки ладьей для третьего теста из примера:



Красным отмечены клетки, которые бьет левая нижняя ладья, синим отмечены клетки, которые бьет правая верхняя, фиолетовым — клетки, которые бьют обе ладьи.

Задача B. Угадай массив

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Это интерактивная задача. Ваша программа будет взаимодействовать с программой жюри, используя стандартный ввод и вывод.

Алиса и Боб решили сыграть в игру под названием «Угадай массив». Правила игры очень простые: Алиса загадала массив целых чисел размера n, а Бобу нужно угадать этот массив, сделав не больше n запросов о сумме на отрезке.

За один ход Боб может сделать к Алисе запрос одного из двух типов:

1. «`? l r`» — узнать сумму чисел на отрезке массива с l-го по r-й элемент включительно;
2. «`!`» — сообщить Алисе, что он готов дать ответ. После этого запроса Алиса ожидает от Боба n целых чисел — загаданный массив.

На каждый запрос первого типа, Алиса говорит Бобу сумму чисел на запрошенном отрезке. Но чтобы отгадывать было сложнее, после каждого запроса Алиса делает один отрезок *запрещенным*. В дальнейших запросах Боб не может запрашивать сумму чисел на запрещенных отрезках.

Помогите Бобу угадать загаданный массив, сделав не более n запросов первого типа.

Протокол взаимодействия

В первой строке ввода дано целое число n — размер загаданного массива ($1 \leqslant n \leqslant 10^4$). Гарантируется, что все числа в массиве по модулю не превосходят 10^9.

Далее запускается протокол взаимодействия с программой жюри — интерактором.

Интерактор ожидает от вашей программы запросы двух типов: «`? l r`» или «`!`», где l, r — целые числа — границы отрезка, на котором вы хотите узнать сумму ($1 \leqslant l \leqslant r \leqslant n$). После каждого запроса должен следовать перевод строки. При несоблюдении вашей программой формата запросов, ваше решение может получить произвольный вердикт (отличный от `OK`).

После запроса первого типа необходимо считать со стандартного ввода три целых числа s, l_b и r_b — сумму чисел на запрошенном отрезке s и границы нового *запрещенного* отрезка $[l_b, r_b]$, на котором запрашивать сумму больше нельзя ($|s| \leqslant 10^{18}$; $1 \leqslant l_b \leqslant r_b \leqslant n$).

Запрос второго типа означает, что ваша программа готова дать ответ на задачу. После запроса второго типа необходимо вывести n целых чисел — загаданный массив.

Обратите внимание, ваша программа может сделать не более n запросов первого типа. При превышении данного ограничения, а также при попытке узнать сумму на *запрещенном* ранее отрезке, интерактор выведет «`-1 -1 -1`» и завершится с вердиктом `WA`. Чтобы не получить вердикт `TL` или `IL`, считав из ввода значения «`-1 -1 -1`», ваша программа должна завершить свою работу с нулевым кодом возврата.

Пример

стандартный ввод	стандартный вывод
3 1 2 2 3 2 3 6 1 2	 ? 1 1 ? 3 3 ? 1 3 ! 1 2 3

Задача C. Перемещение клеток

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Маленькой девочке Алисе на день рождения подарили современную пиксельную картину.

Картина представляет собой клетчатую прямоугольную таблицу размером $n \times m$. Известно, что в каждом столбце этой таблицы несколько клеток подряд покрашены в черный цвет, а все остальные клетки — белые.

Алиса считает картину *красивой*, если существует путь между любыми двумя черными клетками u и v, который проходит только по черным клеткам, каждый раз переходя из клетки в соседнюю с ней по стороне — начать движение в черной клетке u, затем перейти в соседнюю с u по стороне черную клетку w, затем перейти в соседнюю с w по стороне черную клетку, и так далее, добравшись в итоге до черной клетки v.

Так как картина современная, её можно менять. За одно *действие* разрешается выбрать на ней любой столбец и сдвинуть все черные клетки в этом столбце на одну клетку в одном и том же направлении — вверх или вниз. Клетки можно перемещать, только если они не выходят за границы картины.

Алиса заинтересовалась, какое минимальное количество действий необходимо сделать, чтобы картина стала *красивой*.

Формат входных данных

В первой строке ввода даны два целых числа n и m — количество строк и количество столбцов у клетчатой картины соответственно ($1 \leqslant n, m \leqslant 100\,000$). Гарантируется, что площадь картины не превышает 10^6 ($1 \leqslant n \cdot m \leqslant 1\,000\,000$).

В следующих m строках записаны по два целых числа s_i и t_i — номера начальной и конечной позиции непрерывного отрезка черных клеток в i-м столбце клетчатой картины ($1 \leqslant s_i \leqslant t_i \leqslant n$).

Формат выходных данных

Выведите единственное целое число — минимальное количество действий, которое необходимо сделать с картиной, сделать её *красивой*.

Пример

стандартный ввод	стандартный вывод
9 3 1 2 4 5 7 9	4

Задача D. Отряд клонов

Ограничение по времени: 1.5 секунд
Ограничение по памяти: 512 мегабайт

Отряд из x клонов пробрался на корабль «Звезда смерти», чтобы помочь Люку Скайуокеру в битве с Дартом Вейдером. Корабль состоит из n помещений и m двунаправленных переходов между ними. Клоны оказались в помещении 1 и планируют пробраться в помещение n, где находится Люк.

Однако каждое помещение охраняется дроидами, i-е помещение охраняют a_i дроидов. При появлении в помещении повстанцев происходит сражение. Если численность отряда клонов больше численности отряда дроидов, то все дроиды будут убиты, а клоны не понесут потерь. В противном случае клоны также уничтожат всех дроидов, но потеряют половину состава: если в момент начала сражения было x клонов, то в конце останется $\left\lfloor \frac{x}{2} \right\rfloor$, округление вниз. Клонам придется сразиться с дроидами во всех помещениях, в которых они побывают, включая помещения 1 и n.

Помогите командиру отряда понять, какое максимальное количество клонов сможет добраться из помещения 1 до помещения n.

Формат входных данных

В первой строке заданы два целых числа n и m — количество помещений и количество переходов на «Звезде смерти» ($1 \leqslant n, m \leqslant 2 \cdot 10^5$).

В следующих m строках описаны переходы: i-й переход задаётся двумя целыми числами u_i и v_i — номерами помещений, которые соединяет переход ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$). Гарантируется, что каждая пара помещений соединена не более чем одним переходом.

На следующей строке записано целое число x — численность отряда клонов, пробравшихся на корабль ($1 \leqslant x \leqslant 10^9$).

В последней строке заданы n целых чисел $a_1, a_2, \ldots, a_n$ — численности отрядов дроидов в помещениях ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите единственное число — максимальное количество клонов, которые могут добраться до помещения n, начав движение из помещения 1. Если не существует маршрута, при движении по которому в живых останется хотя бы один клон, выведите 0.

Примеры

стандартный ввод	стандартный вывод
4 4 1 2 1 3 2 4 3 4 7 10 2 3 1	3
4 4 1 2 1 3 2 4 3 4 7 10 3 3 1	0

Задача E. Хокку

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Хокку — жанр японской поэзии, представляющий собой трёхстишье, состоящее из 17 слогов, первые 5 из которых находятся в первой строке, следующие 7 находятся во второй строке, и последние 5 находятся в последней строке.

Вы нашли большой английский текст, посвящённый хокку. Однако в нём не оказалось переносов строк. Вы уже разбили текст на слова и теперь хотите найти в нем все потенциальные хокку: отрезки подряд идущих слов, которые могут образовать хокку.

Для простоты в этой задаче приняты следующие соглашения. Словом является последовательность из строчных букв английского алфавита. Гласными буквами считаются буквы «`a`», «`e`», «`i`», «`o`» и «`u`». Количество слогов в слове определяется как количество блоков подряд идущих гласных букв. Например, в слове «`contest`» два слога, а в слове «`beautiful`» — три слога.

Необходимо найти количество отрезков подряд идущих слов, которые, если в них добавить два переноса строки после каких-либо двух слов, будут образовывать хокку.

Например, в тексте «if the real beauties of sunset in a suspended moment call for the thunder forever» встречаются два потенциальных хокку:

the real beauties of
sunset in a suspended
moment call for the

и

beauties of sunset
in a suspended moment
call for the thunder

Формат входных данных

В первой строке входных данных находится число n — количество слов в тексте ($1 \leqslant n \leqslant 10^5$). В следующих n строках находятся слова, состоящие из строчных латинских букв. Длина каждого слова не превышает 20. Гарантируется, что в каждом слове есть хотя бы один слог.

Формат выходных данных

Выведите единственное число — количество потенциальных хокку.

Пример

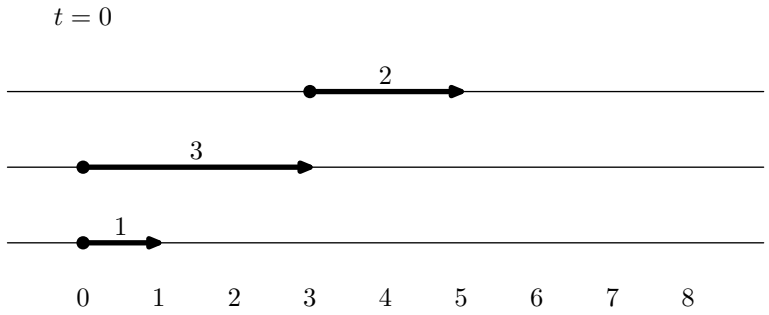
стандартный ввод	стандартный вывод
15 if the real beauties of sunset in a suspended moment call for the thunder forever	2

Задача F. Забег

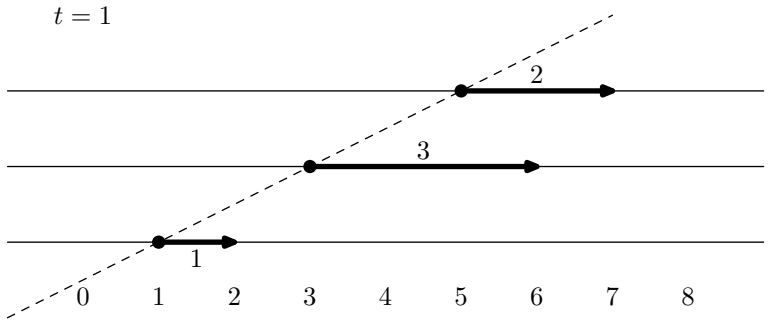
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В забеге участвует n человек, которые бегут по n прямым дорожкам стадиона. Будем считать дорожки горизонтальными прямыми, i-я дорожка представляет собой прямую $y = i$.

Спортсмен номер i начинает в точке (s_i, i) и бежит строго направо со скоростью v_i. Гонка длинная, поэтому можно считать, что спортсмены никогда не останавливаются.



Наблюдающий за забегом начинающий фотограф Даниил заинтересовался, какое максимальное количество спортсменов в какой-либо момент окажутся на одной прямой. Помогите ему это выяснить.



Формат входных данных

В первой строке входных данных содержится целое число n — количество участников забега ($1 \leqslant n \leqslant 300$).

Далее следует n строк, i-я из которых содержит два целых числа s_i и v_i — изначальную x-координату участника с номером i и его скорость, соответственно ($-10^6 \leqslant s_i \leqslant 10^6$; $1 \leqslant v_i \leqslant 10^6$).

Формат выходных данных

Выведите одно число — максимальное количество людей, которые будут находиться на одной прямой в какой-либо момент гонки.

Пример

стандартный ввод	стандартный вывод
3 0 1 0 3 3 2	3

Задача G. Максимизировать сумму XOR

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Будем обозначать как $\oplus$ операцию *побитового «исключающего или»* для целых чисел. В языках программирования C++ и Java она обозначается символом «`^`», в паскале и Python — ключевым словом «`xor`». Например, $9 \oplus 3 = 1001_2 \oplus 11_2 = 1010_2 = 10$.

Даны два массива A и B длины n. Обозначим как $X(A)$ для массива A результат вычисления побитового «исключающего или» от всех элеметов массива: $X(A) = A_1 \oplus A_2 \oplus \ldots \oplus A_n$. Аналогично, введем обозначение $X(B) = B_1 \oplus B_2 \oplus \ldots \oplus B_n$.

Для каждого i от 1 до n разрешается поменять местами элементы A_i и B_i. Необходимо определить, какие из этих обменов надо сделать, чтобы максимизировать сумму $X(A) + X(B)$.

Формат входных данных

В первой строке входных данных находится число n — количество элементов ($1 \leqslant n \leqslant 10^5$). В следующей строке находится n элементов массива A ($0 \leqslant A_i \leqslant 10^{18}$). В следующей строке в таком же формате дан массив B.

Формат выходных данных

В первой строке выведите максимальную возможную сумму и число k — количество необходимых обменов. В следующей строке выведите k различных чисел от 1 до n — индексы элементов, которые надо поменять.

Пример

стандартный ввод	стандартный вывод
2 1 1 2 2	6 1 1

Замечание

В примере после обмена массивы равны $A = [2, 1]$ и $B = [1, 2]$, соответственно.
$X(A) = 2 \oplus 1 = 10_2 \oplus 1_2 = 11_2 = 3$, $X(B) = 3$, $X(A) + X(B) = 6$.

Задача H. Игра в осьминога

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

В некоторой таинственной стране M каждый год проводится турнир по «Игре в осьминога».

Во время очередного раунда игрокам предстоит решить математическую головоломку. Каждому игроку выдаются две карточки, на которых исходно записаны целые числа a_0 и b_0, соответственно.

В процессе игры участники могут совершать действия со своими карточками. Пусть на карточках игрока записаны числа a и b. В качестве действия игрок выбирает целое число k совершает одну из следующих операций:

1. заменить число на первой карточке на $a + kb$;
2. заменить число на второй карточке на $b + ka$.

В процессе игры модуль числа на карточке не должен превышать 10^{18}, иначе с игроком может случиться непоправимое. В раунде побеждают те игроки, которые, совершив не более 50 действий, добиваются того, что на одной из карточек записано число 0.

Вам предстоит сыграть в эту игру и вы конечно же хотите победить!

Формат входных данных

В единственной строке через пробел даны два целых числа a_0 и b_0 — исходные числа на карточках ($-10^{18} \leqslant a_0, b_0 \leqslant 10^{18}$).

Формат выходных данных

В первой строке выведите число n — количество действий, после которых на одной из карточек окажется число 0 ($0 \leqslant n \leqslant 50$). Обратите внимание, что вы не должны минимизировать число действий, но оно не должно превышать 50.

В следующих n строках выведите по два числа через пробел t_i и k_i — информацию о каждом действии: тип действия и выбранное число k.

Если решений несколько, выведите любое, но обратите внимание, что в процессе игры числа по модулю не должны превышать 10^{18}.

Примеры

стандартный ввод	стандартный вывод
-3 9	1 2 3
-27 57	2 2 2 1 9
56 15	6 1 -2 1 -1 2 -2 1 1 2 2 1 -4

Замечание

В первом тесте достаточно сделать одно действие: добавить к числу на второй карточке утроенное число на первой.

Во втором тесте после первого действия на первой и второй карточках записаны числа -27 и 3 соответственно, после второго числа 0 и 3.

В третьем тесте по ходу игры на первой и второй карточках будут записаны соответственно числа 56 и 15, 26 и 15, 11 и 15, 11 и -7, 4 и -7, 4 и 1, 0 и 1.

Задача I. Зачет в третьей параллели

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Один преподаватель придумал новый формат проведения зачета.

- Зачет состоит из n блоков, каждый из которых соответствует одной из пройденных тем; за i-й блок для всех i от 1 до n независимо от других ставится оценка c_i;
- Оценка за каждый блок может принимать любое целое значение от 0 до 100 включительно, и может быть получена на выбор ученика либо ответом на *теоретический вопрос*, либо решением *практической задачи*;
- Зачет считается успешно сданным, если хотя бы a блоков были сданы ответом на теоретический вопрос и хотя бы b блоков были сданы решением практической задачи;
- При соблюдении предыдущего условия итоговая оценка за зачет C вычисляется как сумма оценок за все блоки, то есть $C = \sum_{i=1}^{n} c_i$.

Илья собирается сдавать зачет. Илья достаточно хорошо представляет уровень своих знаний по каждой теме, и уверен, что сдавая i-й блок теорией, получит оценку x_i, а сдавая его же практикой — оценку y_i. Помогите ему определить, какие из блоков (не менее a) ему следует сдавать теорией, а какие (не менее b) — практикой, чтобы набрать максимально возможный суммарный балл за зачет.

Формат входных данных

В первой строке входных данных через пробел перечислены три целых числа n, a и b — общее количество тем, а также минимальное количество тем, которые необходимо сдать теорией и практикой, соответственно ($1 \leqslant n \leqslant 2 \cdot 10^5$; $0 \leqslant a, b \leqslant n$). Гарантируется, что $a + b \leqslant n$.

Во второй строке через пробел перечислены n целых чисел x_i — оценки, которые получит Илья, если будет сдавать блоки, отвечая на теоретические вопросы ($0 \leqslant x_i \leqslant 100$).

В третьей строке так же перечислены n целых чисел y_i — оценки, которые он получит, решая практические задачи ($0 \leqslant y_i \leqslant 100$).

Формат выходных данных

В первой строке выведите одно целое число C — максимальную суммарную оценку, которую Илья может получить за зачет.

Во второй строке выведите n разделеленных пробелами символов, i-й из которых равен «T», если Илье стоит отвечать в i-м блоке теорию, и «P», если практику. Хотя бы a символов должны быть равны «T», и хотя бы b равны «P».

Примеры

стандартный ввод	стандартный вывод
4 1 1 10 30 50 70 80 60 40 20	260 P P T T
4 1 1 30 40 60 90 10 25 50 85	215 T T T P
4 2 1 0 17 70 13 2 21 55 99	190 T P T P

Задача J. Вычислительная этнография

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Аборигены острова Л записывают числа наоборот: старшие разряды в конце числа. Например, число 144 они записывают как 441.

Начинающий этнограф-математик Петя изучает полные квадраты и культуру аборигенов острова Л. Он заметил, что некоторые числа являются полными квадратами и если их рассматривать как обычные, и если их рассматривать как числа, записанные аборигенами острова Л. Например, таким является упомянутое выше число 144, если его считать записанным обычным образом, то $144 = 12^2$, а если считать, что это записанное аборигенами число 441, то $441 = 21^2$. Петя называет такие числа *интересными*.

Петя заинтересовался, сколько существует интересных чисел, лежащих от A до B, включительно.

Формат входных данных

На первой строке ввода записано целое число A, на второй строке ввода записано целое число B ($1 \leqslant A \leqslant B \leqslant 10^{11}$).

Формат выходных данных

Выведите искомое количество интересных чисел.

Пример

стандартный ввод	стандартный вывод
1 1000	10

Замечание

В примере интересными являются числа 1, 4, 9, 121, 144, 169, 441, 484, 676 и 961. Аборигены не используют при записи ведущие нули, поэтому число 100, например, интересным не является.

Задача K. Работай или Спи!

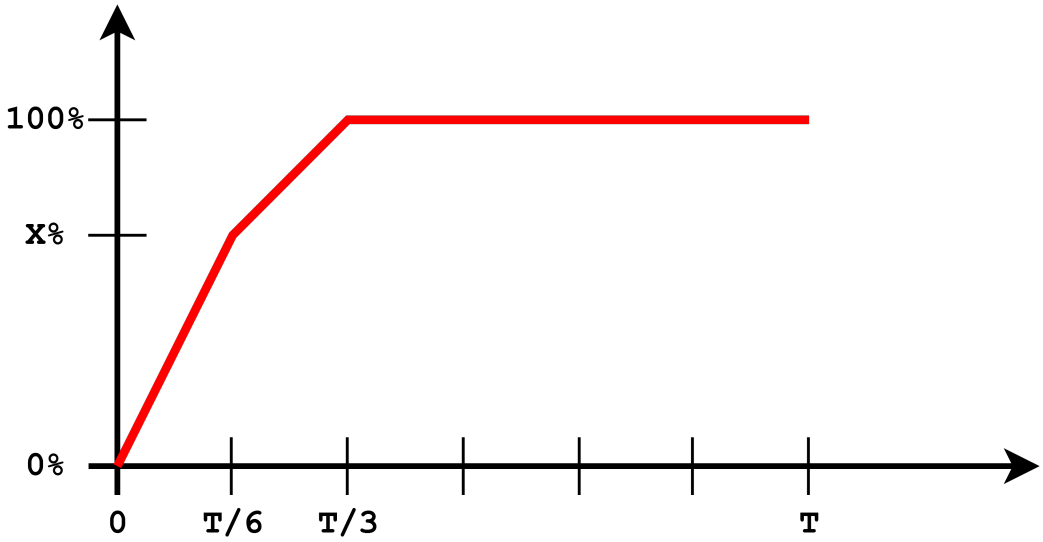
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Олег — программист с планеты, на которой в сутках содержится ровно T часов. Олег придерживается принципа «Работай или Спи». Согласно этому принципу все время, что ты не спишь, ты работаешь.

Недавно Олег на работе прочитал в научном журнале статью про зависимость эффективности программиста в зависимости от времени сна. Результаты исследования показали:

- Если спать $T/3$ часов в сутки, то эффективность будет равна 100%.
- Если спать $T/6$ часов в сутки, то эффективность будет равна X%
- Если спать 0 часов в сутки, то эффективность будет равна 0%
- При этом если спать от $T/6$ до $T/3$ часов в сутки, то эффективность будет линейно возрастать от X% до 100%.
- При этом если спать от $T/6$ до 0 часов в сутки, то эффективность будет линейно падать от X% до 0%.

Олег решил структурировать эту информацию в виде графика функции зависимости эффективности от времени сна и получил следующий результат:



Олег считает, что объем работы, которую он будет выполнять в сутки равен произведению времени работы на эффективность. Соответственно, проблема в том, что больше спишь — меньше работаешь, а меньше спишь – меньше эффективность.

Олег хочет как можно быстрее вернутся к работе, помогите ему определить максимально возможный суточный объем работы, который он сможет выполнить при оптимальном выборе времени сна.

Формат входных данных

В единственной строке заданы два целых числа X и T — эффективность в процентах при времени сна $T/6$ и число часов в сутках на планете Олега, соответственно ($0 \leqslant X \leqslant 100$, $1 \leqslant T \leqslant 10^5$).

Формат выходных данных

Выведите одно число — максимальный объем работы, который Олег сможет выполнить в сутки, с относительной или абсолютной погрешностью не более 10^{-6}.

Примеры

стандартный ввод	стандартный вывод
75 24	1600.00000000
100 24	2000.00000000
77 123	8214.26086957

Замечание

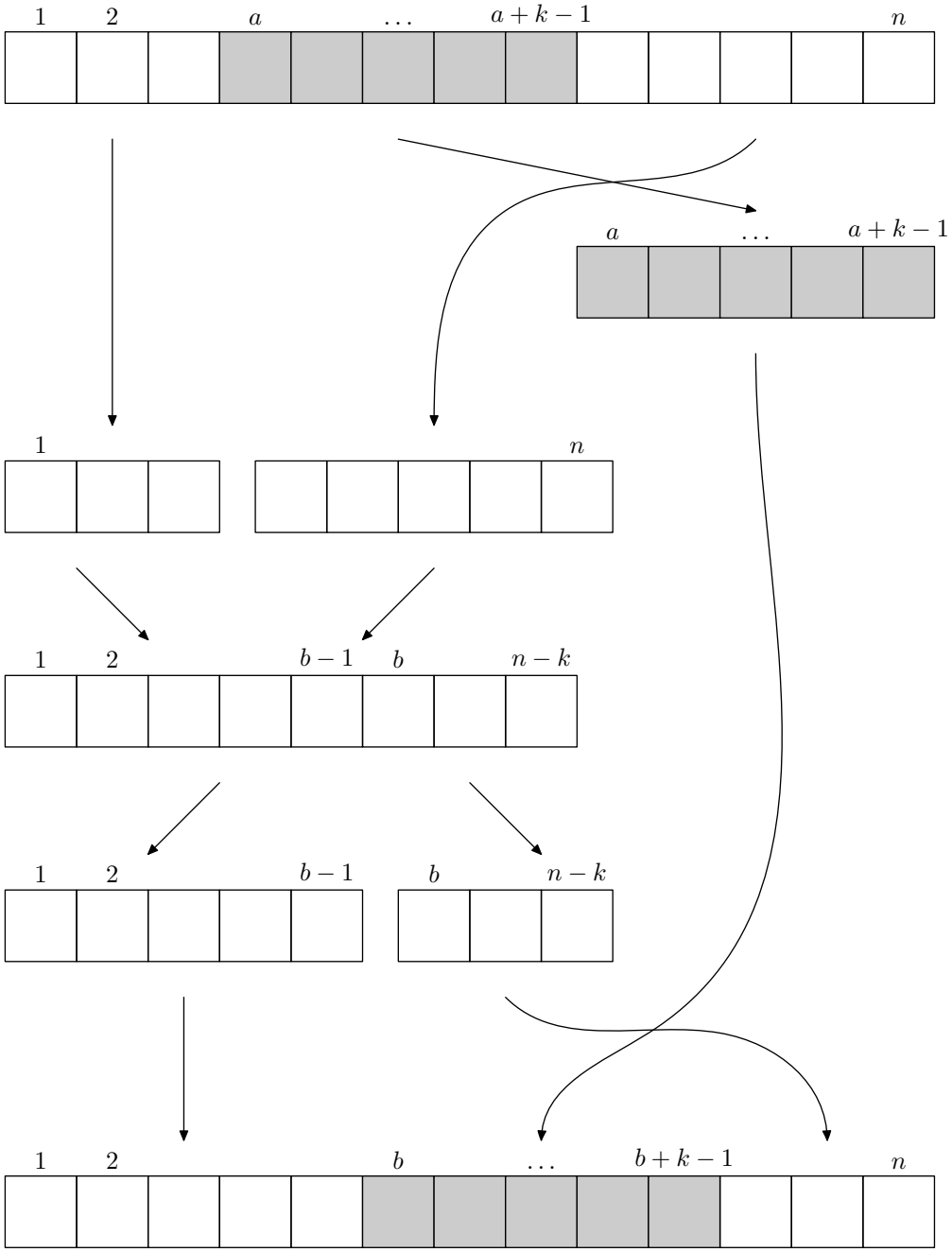
В первом тесте из примера наибольший объем работы Олег сможет выполнить, если будет спать восемь часов. Во втором тесте Олегу нужно спать четыре часа, чтобы добиться максимального объема работы.

Задача L. Трансформация перестановки

Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Федор работает в отделе трансформации перестановок. Сегодня Федор должен решать такую задачу: нужно из перестановки $[p_1, p_2, \ldots, p_n]$ целых чисел $1, 2, \ldots, n$ получить перестановку $[q_1, q_2, \ldots, q_n]$, используя для преобразования не более n^3 операций k-переноса.

Пусть задан массив длиной n. Операция k-переноса с параметрами (a, b) определяется следующим образом: отрезок из k подряд идущих элементов массива, начинающийся с элемента с индексом a, вырезается из массива и вставляется обратно, начиная с индекса b.



Более формально: пусть задан массив $[t_1, t_2, \ldots, t_n]$ и два числа a и b $(1 \leqslant a, b \leqslant n - k + 1)$. Рассмотрим вспомогательный массив $[r_1, r_2, \ldots, r_{n-k}]$, который состоит из чисел $[t_1, t_2, \ldots, t_{a-1}, t_{a+k}, t_{a+k+1}, \ldots, t_n]$. Тогда результатом k-переноса с параметрами (a, b) для массива t называется массив, состоящий из чисел $[r_1, r_2, \ldots, r_{b-1}, t_a, t_{a+1}, \ldots, t_{a+k-1}, r_b, r_{b+1}, \ldots, r_{n-k}]$.

Фёдор не знает, как подступиться к новой задаче, поэтому просит вас помочь ему в этом непростом деле!

Вам необходимо решить задачу для t наборов входных данных.

Формат входных данных

Первая строка содержит одно целое число t ($1 \leqslant t \leqslant 100$) — количество наборов входных данных.

Каждый из наборов входных данных описывается в трёх строках. Первая строка содержит целые числа n и k ($1 \leqslant k \leqslant n \leqslant 100$).

Вторая строка содержит n различных целых чисел $p_1, p_2, \ldots, p_n$ ($1 \leqslant p_i \leqslant n$) — перестановку p.

Третья строка содержит n различных целых чисел $q_1, q_2, \ldots, q_n$ ($1 \leqslant q_i \leqslant n$) — перестановку q.

Гарантируется, что сумма n по всем наборам входных данных не превосходит 100.

Формат выходных данных

Выведите ответы для каждого набора входных данных. Формат ответа для одного набора входных данных описан ниже.

Если из перестановки $p_1, p_2, \ldots, p_n$ невозможно получить перестановку $q_1, q_2, \ldots, q_n$ с помощью k-переносов, требуется вывести «NO» в единственной строке вывода.

Иначе первой строкой вывода должно быть слово «YES».

Во второй строке вывода должно находиться единственное число m — количество выполненных k-переносов для получения перестановки q из перестановки p ($0 \leqslant m \leqslant n^3$). Обратите внимание, что вам не требуется минимизировать число m. Гарантируется, что если перестановку q возможно получить из перестановки p с помощью k-переносов, то существует решение, требующее не более, чем n^3 действий.

В каждой из следующих m строк требуется вывести параметры a и b для очередного k-переноса.

Пример

стандартный ввод	стандартный вывод
3 2 1 2 1 2 1 4 2 1 2 3 4 1 2 4 3 3 2 2 1 3 1 3 2	YES 0 NO YES 2 1 2 1 2

Замечание

В третьем наборе входных данных перестановку q из перестановки p можно получить и другим способом — использовав один k-перенос с параметрами $a = 2$, $b = 1$.