

Задача А. Треугольники

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Любимая геометрическая фигура дочерей Грю — треугольник. В свободное от занятий и помощи Грю в его злодействах время девочки любят играть в следующую игру: каждая из них выбирает целое положительное число и после этого они вместе проверяют, может ли существовать невырожденный треугольник со сторонами, имеющими длины, равные выбранным числам.

В какой-то момент нашли дома набор из n положительных целых чисел a . Они хотят узнать, сколько существует целых x таких, что x и любые два другие числа из набора a всегда будут являться длинами сторон некоторого невырожденного треугольника. Помогите девочкам решить эту задачу.

Формат входных данных

В первой строке входных данных дано одно целое число — n ($2 \leq n \leq 5 \cdot 10^5$).

Во второй строке входных данных дано n целых чисел — числа набора a ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Выведите одно целое число — ответ на вопрос задачи.

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	Информация о проверке
0	–	примеры из условия		полная
1	14	$n \leq 500$; $a_i \leq 200$ для всех i	0	полная
2	15	$n \leq 2000$; $a_i \leq 500$ для всех i	0, 1	первая ошибка
3	11	$a_i \leq 4$ для всех i		первая ошибка
4	13	$a_i = a_j$ для всех i и j		первая ошибка
5	14	$a_1 = 1$		первая ошибка
6	16	$a_i \leq 2 \cdot 10^5$ для всех i	0, 1, 2, 3	первая ошибка
7	17	без дополнительных ограничений	0 – 6	первая ошибка

Примеры

стандартный ввод	стандартный вывод
3 3 3 5	3
3 3 1 2	0
5 9 5 6 7 9	6

Задача В. Взлом сейфа

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Грю необходимо взломать сейф Вектора, в котором находятся чертежи ракеты, специально разработанной для миссии по похищению Луны. Разумеется, это не очень простая задача, поэтому поручать ее Миньонам не стоит. Поэтому решить ее предстоит вам.

Кодовый замок имеет экран, и на этом экране сейчас написана строка s , состоящая из маленьких букв английского алфавита. Любимое слово Вектора — строка t той же длины, что и строка s . Грю уверен, что замок откроется, когда на экране вместо s будет отображаться строка t .

Рядом с экраном есть поле для ввода целого числа и одна единственная кнопка. Доктор Нефарио проанализировал это устройство и сообщил Грю и вам, что в поле можно ввести любое целое число от 1 до $|s|$, а при нажатии кнопки

1. строка s разбивается на префикс p длины $|s| - x$ (строку из первых $|s| - x$ символов s) и суффикс q длины x (строку из оставшихся x символов), где x — введенное в поле число;
2. и затем s заменяется на $q^{\text{rev}}p$, то есть на конкатенацию (запись подряд) развернутой строки q и строки p .

Иными словами, суффикс строки s длины x разворачивается и переставляется в начало строки. Например, ввод числа 2 и нажатие на кнопку заменит строку «arkshs» на «sharks».

Логично, что слишком большое количество нажатий на кнопку вызовет определенные подозрения, и увеличит ваши шансы быть пойманными, поэтому количество применений описанной операции ограничено. Найдите способ получить на экране строку t , используя не более чем m операций.

Формат входных данных

Первая строка ввода содержит два целых числа n и m — длины строк s и t , а также максимальное число операций ($1 \leq n \leq 2000$; $5100 \leq m \leq 10^4$).

Во второй и третьей строках ввода даны s и t , состоящие из n маленьких букв английского алфавита (символы от 'a' до 'z').

Формат выходных данных

Если нельзя получить t из s , используя не более m нажатий на кнопку, выведите единственное число «-1». Иначе в первой строке выведите количество нажатий на кнопку k ($0 \leq k \leq m$), а в следующей строке выведите через пробел k целых чисел, i -е из которых равно числу, которое должно быть введено перед i -м нажатием кнопки.

Обратите внимание, что минимизировать количество нажатий на кнопку не требуется — достаточно просто уложиться в определенные ограничения.

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	Информация о проверке
0	–	примеры из условия		полная
1	9	$n \leq 8, m = 10^4$	0	полная
2	21	$n \leq 100, m = 10^4$	0, 1	первая ошибка
3	24	$n \leq 1000, m = 10^4$	0 – 2	первая ошибка
4	12	$m = 10^4$	0 – 3	первая ошибка
5	12	$m \geq 8100$	0 – 4	первая ошибка
6	11	$m \geq 6100$	0 – 5	первая ошибка
7	11	без дополнительных ограничений	0 – 6	первая ошибка

Примеры

стандартный ввод	стандартный вывод
6 10000 abacbb babcba	4 6 3 2 3
3 10000 aba bba	-1

Замечания

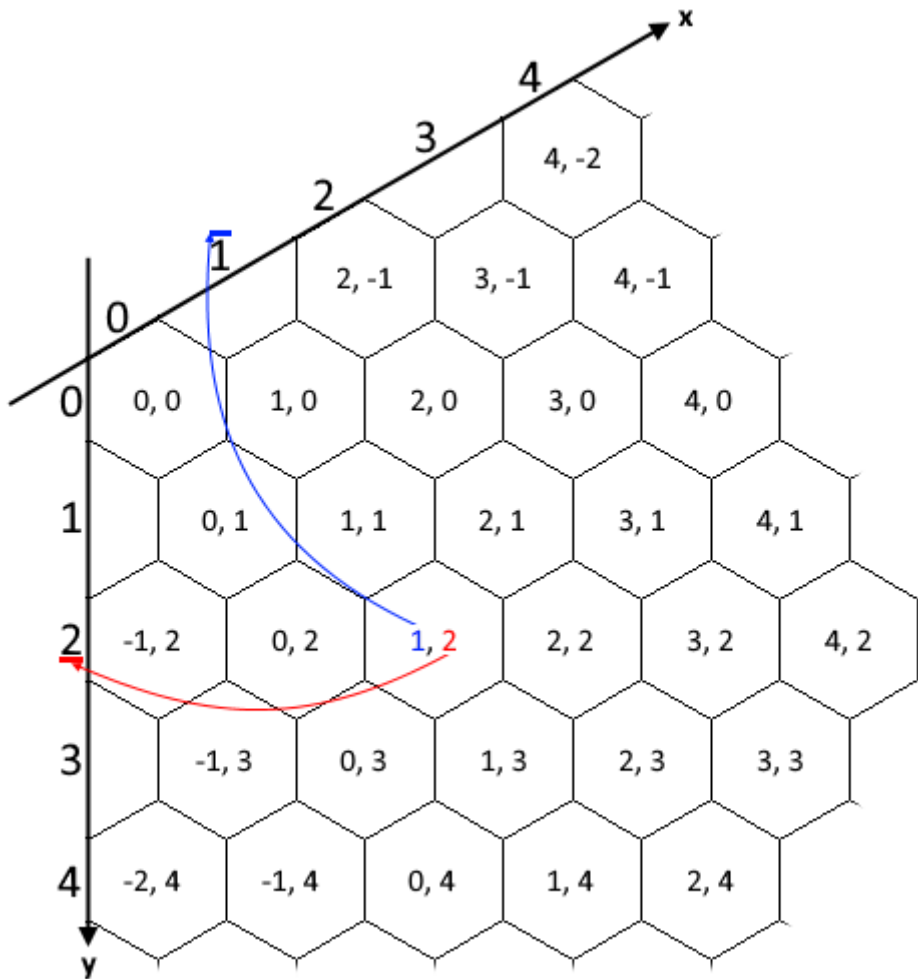
В первом примере после применения операций строка на экране будет изменяться следующим образом:

1. abacbb → bbcaba
2. bbcaba → ababbc
3. ababbc → cbabab
4. cbabab → babcba

Задача С. Шестиугольный рисунок

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Кевин (один из миньонов) увлекся рисованием на шестиугольных решетках. *Шестиугольной решеткой* называется решетка, разбивающая плоскость на равные правильные шестиугольники. Каждый шестиугольник однозначно соответствует паре целочисленных координат, где первая ось координат направлена строго вниз, а вторая — под углом 120° к ней. Рисунок, изображающий решетку и систему координат, можно видеть ниже:



Разумеется, решетка продолжается бесконечно во все стороны, но для удобства изображена только ее часть. Внутри каждой ячейки написаны ее координаты. Для ячейки $(1, 2)$ стрелками показано, как найти ее координаты на осях координат.

Две ячейки называются *соседними*, если их границы имеют общую сторону. Например, ячейки $(1, 3)$ и $(2, 2)$ — соседние, а $(1, 3)$ и $(3, 2)$ — нет. Множество ячеек называется *связным*, если от любой ячейки множества до любой можно «добраться», двигаясь каждый раз в ячейку из множества, соседнюю с предыдущей, и *двусвязным*, если между любыми двумя ячейками множества существует хотя бы два непересекающихся по ячейкам пути.

Сегодня Стюарт и Боб закрасили некоторое (не обязательно связное) множество из n ячеек этой решетки, и показали Кевину. Начинаящий художник считает множество *красивым*, если в нем нет трех взаимно соседних ячеек (имеющих общую точку на границах). Будем называть три такие ячейки *треугольным скоплением*.

Теперь Кевину интересно, какие ячейки следует стереть с рисунка, чтобы в нем не осталось тре-

угольных скоплений. Разумеется, некрасивым рисунком он хвастаться Грю не будет, но, чтобы не портить изначальную задумку Стюарта и Боба, ему хочется стереть как можно меньше закрашенных ими ячеек.

Помогите ему с этой задачей; от вас **не требуется** найти минимальное количество ячеек, которое надо удалить, однако чем меньше ячеек удаляется вашим решением, тем больше баллов оно наберет.

Формат входных данных

В первой строке ввода через пробел даны два целых числа n , cost и γ — количество закрашенных ячеек на рисунке, который получил Кевин ($1 \leq n \leq 10^5$), а также стоимость данного теста и нижняя граница на допустимую точность (см. секцию «Система оценки»). Параметры cost и γ служат для оценки вашего ответа чекером и не обязаны использоваться в вашем решении.

В следующих n строках перечислены координаты ячеек, входящих в множество. В i -й строке через пробел даны два целых числа x_i и y_i — координаты i -й ячейки ($|x_i|, |y_i| \leq 10^9$). Гарантируется, что пары (x_i, y_i) уникальны, то есть что никакая ячейка не перечислена дважды.

Формат выходных данных

В первой строке выведите целое число k — количество ячеек, которые вы удаляете. В следующих k строках перечислите координаты удаляемых ячеек в том же формате, что и во входных данных.

Если вы выводите ячейку, не принадлежащую изначальному множеству, или после удаления всех перечисленных вами ячеек в множестве все еще остаются треугольные скопления, ваше решение получает вердикт WA.

Система оценки

В этой задаче всего 43 теста, не считая примеров к условию. Все тесты оцениваются независимо. Если ваш ответ на определенный тест корректен, за этот тест вы получаете

$$\text{score} = \begin{cases} 0 & \text{если } \frac{j}{p} < \gamma \\ \text{cost} \cdot \min\left(1, \frac{j}{p}\right) & \text{иначе} \end{cases}$$

баллов, где cost — стоимость теста, p — количество ячеек, удаляемых в вашем ответе, j — количество ячеек, удаляемых решением жюри, и γ — нижняя граница на «оптимальность ответа».

Иными словами, если ваш ответ более чем в γ^{-1} раз хуже ответа жюри, вы получаете 0 баллов за этот тест. Иначе вы получаете долю от максимального числа баллов за тест, отражающую насколько оптимален ваш ответ в сравнении с ответом жюри.

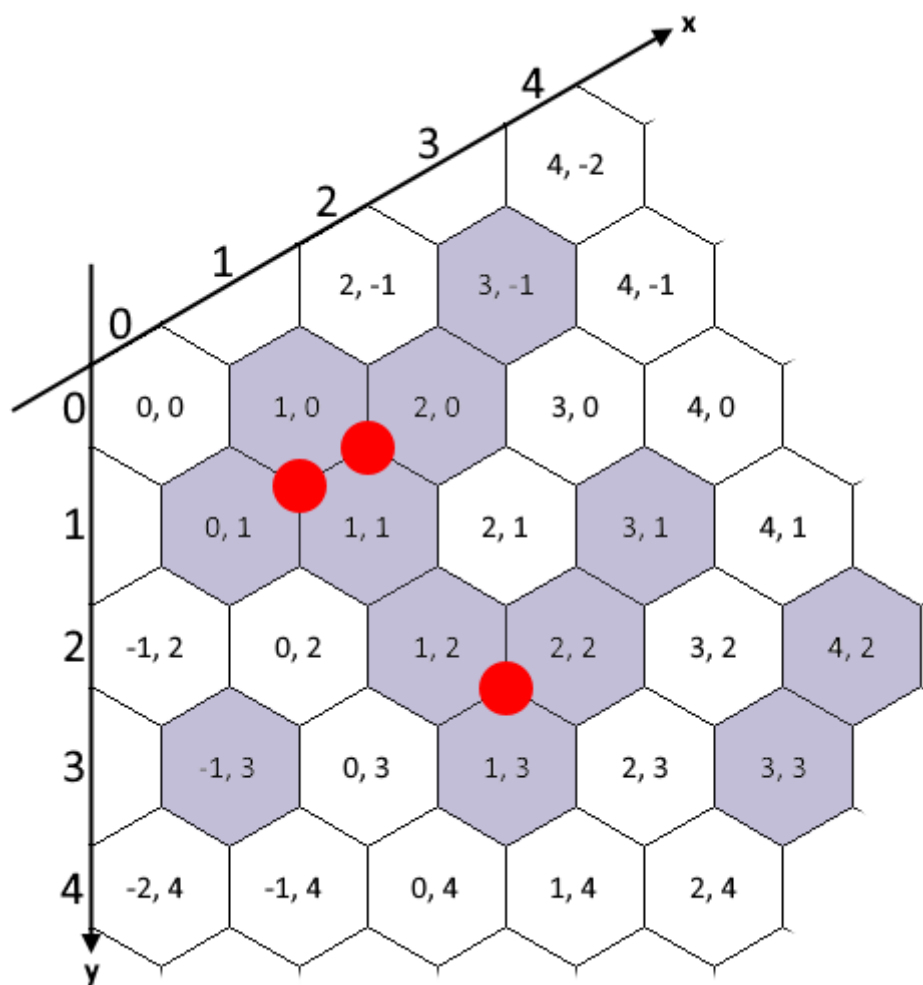
Тесты	Баллы cost	Граница γ	Доп. ограничения
1 – 5	2	0	$n \leq 3$
6 – 10	3	0.5	$n \leq 18$
11 – 20	2	0.25	$n \leq 1000$, множество двусвязно
21 – 30	2	0.7	каждая ячейка множества имеет не более трех соседей
31 – 39	3	0.5	нет
40 – 43	5	1.0	нет

Примеры

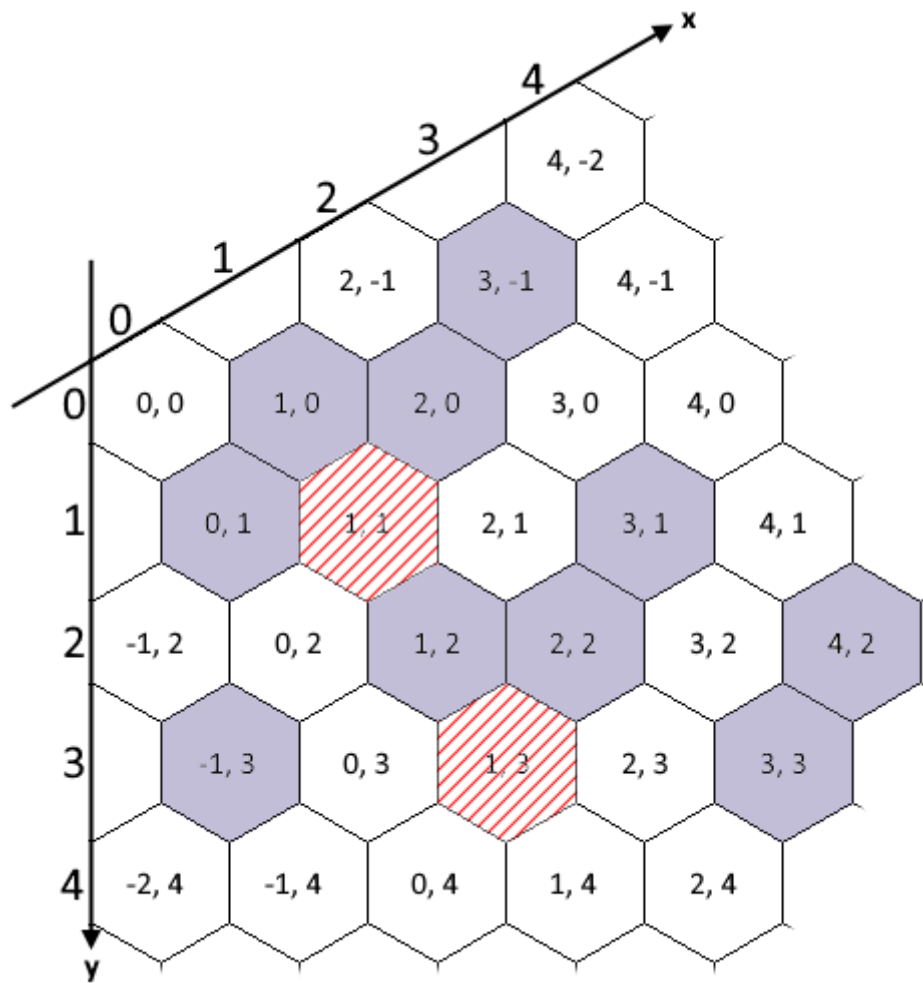
стандартный ввод	стандартный вывод
3 0 0.0 1 0 0 1 1 1	1 1 0
7 0 0.0 1 2 0 2 1 1 2 1 2 2 1 3 0 3	1 1 2
12 0 0.0 1 0 3 -1 0 1 2 0 1 1 -1 3 1 2 3 1 2 2 1 3 4 2 3 3	2 1 1 1 3

Замечание

На рисунках ниже нарисовано множество из третьего примера к условию, а также отмечены все треугольные скопления, и какие ячейки следует удалить.



Все ячейки множества выделены фиолетовым цветом. Красные круги соответствуют треугольным скоплениям. Первое из них образовано ячейками $(0, 1)$, $(1, 0)$ и $(1, 1)$; второе — ячейками $(1, 0)$, $(1, 1)$ и $(2, 0)$; и третье — $(1, 2)$, $(1, 3)$ и $(2, 2)$.



Очевидно, что удаление одной ячейки не позволит избавиться от всех треугольных скоплений. А если удалить две, например, (1, 1) и (1, 3), треугольных скоплений не останется.

Задача D. Путешествие миньонов

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Миньоны много путешествовали в поисках того, кто сможет стать их новым лидером, перед тем, как встретились с Грю. Наиболее перспективными были n городов, i -й из которых, по их оценкам, обладает *перспективностью* a_i .

Разумеется, это не единственный критерий оценки городов. Миньонам также важно весело провести время. При этом посетить город i можно двумя способами:

- либо проездом, в случае чего миньоны получают $(a_i - c)^2$ удовольствия от посещения данного города;
- либо заехав в город на день, и в таком случае миньоны получают $(a_i - a_{\text{prev}})^2$ удовольствия от его посещения, где prev — номер последенного города, в котором миньоны находились в течение дня (не проездом).

При этом в случае, когда миньоны в первый раз за всю поездку заезжают на день в какой-то город (пусть это город i), стоит считать a_{prev} равным a_i , и в таком случае они не получают удовольствия от его посещения.

Миньоны хотят посетить все города последовательно, то есть строго в порядке от 1 до n , при этом, конечно же, они хотят получить от этого максимальное удовольствие. Подскажите им, какой максимальное удовольствие от такого путешествия можно получить.

При этом в города номер 1 и номер n **обязательно заехать на день!** Таким образом, удовольствия от посещения первого города они не получат.

Формат входных данных

В первой строке ввода даны два целых числа n и c — количество городов, которые миньоны хотят посетить, и параметр из условия ($1 \leq n \leq 10^6$; $-10^6 \leq c \leq 10^6$).

Во второй строке перечислены целые числа a_i — значения перспективности городов ($-10^6 \leq a_i \leq 10^6$).

Формат выходных данных

Выведите единственное целое число — максимальное удовольствие, которое миньоны могут получить, посещая города последовательно, начиная с города под номером 1 и **заехав на день** в город 1, и в город n .

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	Информация о проверке
0	–	примеры из условия		полная
1	30	$n \leq 500$	0	первая ошибка
2	20	$n \leq 2000$	0, 1	первая ошибка
3	20	$a_i \leq a_{i+1}$ для всех $i < n$		первая ошибка
4	30	без дополнительных ограничений	0 – 3	первая ошибка

Примеры

стандартный ввод	стандартный вывод
6 3 5 1 6 5 0 1	82
6 -1 4 4 1 1 5 9	138