

Задача 1. Космодром

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

В последнее время все больше марсиан совершают межпланетные перелеты. А в результате развития межпланетной торговли увеличилось количество грузовых космических кораблей, которые прилетают на Марс. Чтобы избежать перегрузки существующих транспортных узлов, мэрия Олимп-Сити решила построить около города новый космодром.

Перед началом работ была подготовлена строительная площадка. Для этого местность около Олимп-Сити разбили на квадраты, образующие сетку. Стороны квадратов параллельны сторонам света. Введем систему координат: обозначим координаты некоторого квадрата как $(0, 0)$, а оси координат направим с севера на юг и с запада на восток соответственно. Тогда квадрат (x, y) граничит по стороне с квадратами $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$ и $(x, y + 1)$.

По плану территория космодрома должна быть квадратной. Т. е. будут выбраны некоторые три целых числа x_0 , y_0 и p ($p \geq 1$). Все квадраты с координатами (x, y) такими, что $x_0 \leq x \leq x_0 + p - 1$ и $y_0 \leq y \leq y_0 + p - 1$ будут находиться на территории космодрома, а остальные — вне его территории.

Квадраты на территории космодрома, которые граничат хотя бы с одним квадратом не на территории космодрома, будем называть *граничными*. Все граничные квадраты должны содержать забор, а на остальных квадратах (как внутри, так и вне космодрома) забора быть не должно.

Строители уже поставили забор на некоторые квадраты. Вам необходимо проконтролировать их работу и убедиться, что построенный забор действительно ограничивает квадратную территорию космодрома. Иными словами, вам необходимо проверить, что забор находится во всех граничных квадратах космодрома с некоторыми параметрами x_0 , y_0 и p , а остальные квадраты не содержат забора.

Формат входных данных

В первой строке входных данных находится целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество квадратов, в которых расположен забор.

В каждой из следующих n строк находится по два целых числа x_i и y_i ($-10^5 \leq x_i, y_i \leq 10^5$) — координаты i -го квадрата, где расположен забор.

Гарантируется, что координаты всех перечисленных во входных данных квадратов различны.

Формат выходных данных

Если описанный во входных данных забор не ограничивает никакой квадратный космодром, выведите одно целое число -1 . Иначе выведите три целых числа x_0 , y_0 , p ($-10^6 \leq x_0, y_0 \leq 10^6$, $1 \leq p \leq 2 \cdot 10^6$) — параметры космодрома, который ограничен построенным забором.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leq 4$	3	
2	$n \leq 1000$	23	1
3	$0 \leq x_i, y_i \leq 1000$	16	
4	Нет дополнительных ограничений	58	1 - 3

Примеры

стандартный ввод	стандартный вывод
12 1 1 1 2 1 3 1 4 2 4 3 4 4 4 4 3 4 2 4 1 3 1 2 1	1 1 4
13 1 1 1 2 1 3 1 4 2 4 3 4 4 4 4 3 4 2 4 1 3 1 2 1 5 2	-1
11 1 1 1 2 1 3 1 4 2 4 4 4 4 3 4 2 4 1 3 1 2 1	-1

Замечание

В первом примере построенный забор выглядит следующим образом:

	(1,1)	(1,2)	(1,3)	(1,4)	
	(2,1)			(2,4)	
	(3,1)			(3,4)	
	(4,1)	(4,2)	(4,3)	(4,4)	

Можно заметить, что если космодром имеет параметры $x_0 = 1$, $y_0 = 1$ и $p = 4$, то тогда забором покрыты все его граничные клетки.

Во втором примере забор выглядит следующим образом:

	(1,1)	(1,2)	(1,3)	(1,4)	
	(2,1)			(2,4)	
	(3,1)			(3,4)	
	(4,1)	(4,2)	(4,3)	(4,4)	
		(5,2)			

Здесь необходимо вывести -1 , поскольку квадрат $(5, 2)$ содержит забор, хотя забором должны быть покрыты только граничные клетки.

Теперь рассмотрим третий пример:

	(1,1)	(1,2)	(1,3)	(1,4)	
	(2,1)			(2,4)	
	(3,1)				
	(4,1)	(4,2)	(4,3)	(4,4)	

Здесь граничный квадрат $(3, 4)$ должен быть покрыт забором, а он является пустым. Поэтому ответ, как и в предыдущем примере, равен -1 .

Задача 2. Игра с числами

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

Мальчик Петя со своим другом, роботом Petya++, любит играть в различные игры. Недавно ребята придумали новую игру и решили сыграть в нее.

Правила игры описаны ниже. В начале задается некоторый набор из n целых чисел a_i . Каждый ход один из игроков может либо поменять знак некоторого числа из набора на противоположный, либо не делать ничего. В конце игры вычисляется сумма всех чисел в полученном наборе. Задача Пети — максимизировать данную сумму, а Petya++ — минимизировать ее.

Особенность данной игры в том, что игроки не обязательно ходят по порядку. Перед началом игры выбирается целое число k — общее количество ходов за игру. Затем игроки k раз тянут жребий, чтобы определить, кто будет ходить на i -м ходу.

Друзья очень долго играли в эту игру, пока им не надоело. Затем они занялись поиском оптимальной стратегии, но так и не сумели ее отыскать. Поэтому ребята обратились за помощью к Вам и попросили написать программу, которая по заданному исходному набору чисел и порядку ходов сможет определить счет игры, если Петя и Petya++ будут играть оптимально.

Формат входных данных

В первой строке входных данных находятся два целых числа n и k ($1 \leq n \leq 3 \cdot 10^5$, $1 \leq k \leq 3 \cdot 10^5$) — количество чисел в наборе и количество ходов в игре.

Во второй строке входных данных находится n целых чисел a_i ($-10^9 \leq a_i \leq 10^9$) — набор, который игроки изменяют во время игры.

В третьей строке входных данных находится k целых чисел b_i ($1 \leq b_i \leq 2$). Если $b_i = 1$, то на i -м ходу ходит Петя, иначе ходит Petya++.

Формат выходных данных

Выведите одно целое число — сумму чисел в наборе после того, как игроки сделают k ходов при оптимальной игре.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	Все b_i равны	3	
2	$n \leq 10$	5	
3	$n \leq 100$, $ a_i \leq 100$	7	
4	$n \leq 1000$	32	2, 3
5	Нет дополнительных ограничений	53	1 - 4

Примеры

стандартный ввод	стандартный вывод
3 2 3 4 5 2 1	12
3 2 3 4 5 1 2	2

Замечание

В первом примере Petya++ может поменять знак на любом из чисел. Следующим ходом Петя меняет знак у этого числа обратно. Тогда набор чисел не изменится, а сумма будет равна $3 + 4 + 5 = 12$.

Во втором примере первым ходит Петя. Ему выгоднее всего не менять никакое из чисел. Следующим ходом Petya++ изменяет число 5 на -5 . В конце игры сумма получается равной $3+4+(-5) = 2$.

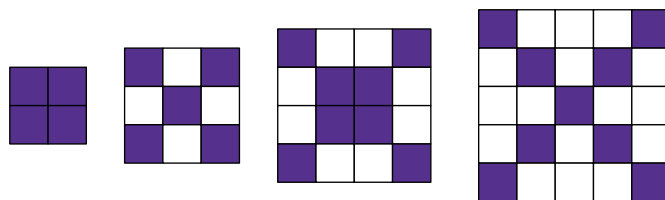
Задача 3. Вражеские шпионы

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

Инопланетные захватчики из одной далекой звездной системы атакуют Марс. Для победы над противником у марсиан есть современная армия и мощное вооружение, поэтому жители Марса не сомневаются в своей победе. Однако если врагу удастся заполучить военные секреты марсиан, то его войска доставят куда больше неприятностей, а победа марсиан будет не столь очевидной. Поэтому генерал марсианской армии поставил перед своими подчиненными крайне важную задачу: найти и уничтожить всех вражеских шпионов.

Для поиска шпионов была произведена аэрофотосъемка местности. В результате получилась черно-белая фотография, которую можно представить в виде матрицы, состоящей из n строк и m столбцов. Каждый элемент матрицы является либо черным, либо белым пикселем.

Известно, что шпионские дроны противника имеют крестообразную форму. Поэтому был разработан следующий алгоритм поиска. Рассматриваются все квадратные подматрицы с длиной стороны, **строго** большей 1. Если в некоторой подматрице все пиксели на двух главных диагоналях черные, а все остальные пиксели белые, то данная подматрица считается *подозрительной*. Примеры подозрительных подматриц приведены на рисунке ниже:



После определения всех подозрительных подматриц необходимо подсчитать их количество, чтобы оценить число шпионов на участке местности.

По итогам аэрофотосъемки накопилось довольно большое количество снимков, и теперь стоит вопрос автоматизированной обработки всех этих изображений. Решить данный вопрос должна известная компания «Interplanetary Software, Inc.».

Вы — сотрудник «Interplanetary Software, Inc.». Вам было поручено разработать программу, которая по заданному снимку определяет количество подозрительных подматриц на нем.

Формат входных данных

В первой строке входных данных находится два целых числа n и m ($1 \leq n, m \leq 2000$) — размеры изображения.

В каждой из следующих n строк находится по одной строке, состоящей из m символов «.» и «#» — описание изображения. Символу «.» соответствует белый пиксель, а символу «#» — черный.

Формат выходных данных

Выведите одно целое число — количество подозрительных подматриц в заданном изображении.

Система оценки

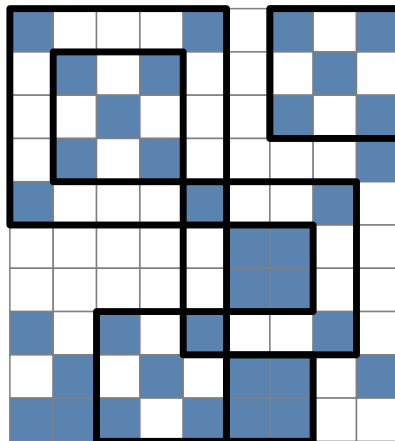
№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n, m \leq 2$	3	
2	$n, m \leq 3$	1	1
3	$n, m \leq 20$	14	1, 2
4	$n, m \leq 300$	21	1 - 3
5	Нет дополнительных ограничений	61	1 - 4

Примеры

стандартный ввод	стандартный вывод
<pre> 10 9 #...#.#.# .#.#...#. ..#...#.# .#.#...#. #...#...###..##.. #.#.#...#. .#.#.##.# ###.###.. </pre>	<pre> 7 </pre>
<pre> 8 8 #.....# .#.....# ..#...#.. ...##... ...##... ..#...#.. .#.....# #.....# </pre>	<pre> 4 </pre>

Замечание

На рисунке ниже изображен первый пример из условия. Все семь подозрительных подматриц выделены:



Задача 4. Мощный процессор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Инженеры одной из крупнейших марсианских корпораций «Black Hole Semiconductors, Inc.» выпустили новый мощный процессор ВН-42. Одной из ключевых особенностей данного процессора является аппаратное ускорение часто используемых алгоритмов. Например, процессор может решать так называемую *задачу Xorseq*, т. е. быстро выполнять следующие операции над массивом a из n чисел:

- Для всех чисел i от l до r поменять значение a_i на $a_i \oplus v$. Здесь \oplus обозначает операцию побитового исключающего ИЛИ.
- Найти длину наибольшей возрастающей подпоследовательности на отрезке $[l; r]$, т. е. найти наибольший по размеру набор чисел (i_1, i_2, \dots, i_k) такой, что $l \leq i_1 < i_2 < \dots < i_k \leq r$ и $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Компания-конкурент, «Interplanetary Software, Inc.» поставляет своим клиентам компьютеры с процессорами Mars-24, которые не умеют эффективно решать описанную выше задачу. Однако инженеры компании считают, что хорошая программная реализация позволит обрабатывать операции Xorseq не медленнее, чем процессор ВН-42.

Теперь перед компанией стоит крайне важная задача: разработать эффективный алгоритм для Xorseq, чтобы убедить клиентов не переходить на процессоры конкурента. Вам, как специалисту по алгоритмам в компании «Interplanetary Software, Inc.» было поручено написать реализацию требуемого алгоритма.

Формат входных данных

В первой строке входных данных находится два целых числа n и q ($1 \leq n \leq 10^5$, $1 \leq q \leq 10^5$) — размер массива и количество операций соответственно.

Во второй строке входных данных находится n целых чисел a_i ($0 \leq a_i \leq 7$) — исходный массив.

В каждой из следующих строк описана ровно одна операция. Описание операции может иметь один из заданных ниже форматов:

- `xor l r v` ($1 \leq l \leq r \leq n$, $0 \leq v \leq 7$) — заменить a_i на $a_i \oplus v$ для всех $l \leq i \leq r$.
- `seq l r` ($1 \leq l \leq r \leq n$) — вывести длину наибольшей возрастающей подпоследовательности на отрезке $[l; r]$.

Формат выходных данных

Для каждой операции второго типа в отдельной строке выведите целое число — ответ на данную операцию. Ответы должны следовать в том же порядке, в котором заданы сами операции.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n \leq 2, q \leq 10$	3	
2	$n \leq 4, q \leq 10$	1	1
3	$n \leq 50, q \leq 50$	7	1, 2
4	$n \leq 1000, q \leq 1000$	11	1 - 3
5	$a_i \leq 1$ в любой момент времени	13	
6	Нет запросов типа <code>xor</code>	9	
7	$n, q \leq 50000$	22	1 - 4
8	Нет дополнительных ограничений	34	1 - 7

Пример

стандартный ввод	стандартный вывод
10 11	4
0 1 4 3 0 7 2 5 2 3	6
seq 1 10	6
xor 3 5 6	1
seq 1 6	2
seq 1 10	5
seq 3 3	2
seq 5 10	1
xor 7 10 5	
seq 2 9	
xor 1 10 7	
seq 2 10	
seq 5 5	

Замечание

Исключающее ИЛИ — операция, которая на вход принимает два бита и возвращает бит 0, если биты на входе равны и 1 в противном случае. Например: $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. Для применения побитового исключающего ИЛИ двух чисел эти числа сначала переводят в двоичную систему счисления, а затем применяют исключающее ИЛИ к каждому из разрядов. Например, $6 \oplus 3 = 5$, поскольку $6 = 110_2$, $3 = 11_2$. Применив исключающее ИЛИ поразрядно, получаем $5 = 101_2$:

$$\begin{array}{r} 110_2 \\ \oplus 11_2 \\ \hline 101_2 \end{array}$$

В языке программирования Pascal побитовое исключающее ИЛИ чисел a и b обозначается $a \text{ xor } b$, а в языках Python и C++ — $a \wedge b$.

Теперь рассмотрим пример из условия.

После первой операции мы выводим ответ 4, поскольку можно выбрать неубывающую подпоследовательность длины четыре: **0 1 4 3 0 7 2 5 2 3**.

После второй операции массив станет выглядеть так: 0 1 2 5 6 7 2 5 2 3.

Ответ на третью операцию равен 6, т. к. отрезок из первых шести чисел имеет вид 0 1 2 5 6 7 и уже является неубывающим.

Для четвертой операции оптимальная подпоследовательность выглядит как **0 1 2 5 6 7 2 5 2 3**.

В пятой операции отрезок состоит из одного элемента, поэтому ответ равен 1.

В шестой операции оптимальная подпоследовательность выглядит как **7 2 5 2 3**.

После седьмой операции массив станет выглядеть как 0 1 2 5 6 7 7 0 7 6.

В восьмой операции оптимальная подпоследовательность выглядит как **1 2 5 6 7 7 0 7**.

После девятой операции массив станет выглядеть как 7 6 5 1 0 0 7 0 1.

В десятой операции оптимальная подпоследовательность выглядит как **6 5 2 1 0 0 7 0 1**.

В одиннадцатой операции отрезок состоит из одного элемента, поэтому ответ равен 1.