

Задача А. Линейный конгруэнтный метод

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 4 мегабайта

У вас есть массив $a_0, a_1, \dots, a_{n-2}, a_{n-1}$. Вам необходимо для каждого i найти такие элементы l_i и r_i ($l_i \leq i \leq r_i$), что для всех k таких, что $l_i \leq k \leq r_i$ выполняется $a_k \leq a_i$, а так же l_i должен быть как можно минимален, а r_i - как можно максимален, а затем найти сумму $r_i - l_i + 1$ для всех i ($0 \leq i \leq n - 1$).

К сожалению, ограничения по памяти в этой памяти очень маленькие, так что вам придется сгенерировать его самим! Вам будет даны длина массива и его первый элемент a_0 . Для всех i , таких что $1 \leq i \leq n - 1$ соблюдается следующее утверждение: $a_i = (a_{i-1} * 214013 + 2531011) \bmod 2^{32}$.

Удачи!

Формат входных данных

В единственной строке заданы n ($3 \leq n \leq 10^7$) и a_0 ($0 \leq a_0 < 2^{32}$).

Формат выходных данных

Выведите ответ на задачу.

Система оценки

Данная задача содержит три подзадачи:

1. $N = 3$. Оценивается в 1 балл.
2. $3 < N \leq 1000$. Оценивается в 29 баллов.
3. $1000 < N \leq 5 \cdot 10^5$. Оценивается в 20 баллов.
4. $5 \cdot 10^5 < N \leq 10^7$. Оценивается в 50 баллов.

Пример

стандартный ввод	стандартный вывод
3 0	6

Замечание

$a \bmod b$ это остаток от деления a на b .

В первом примере массив выглядит как $\{0, 2531011, 505908858\}$.

Задача В. Дробь

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

Эта задача немного необычна — в ней вам предстоит реализовать интерактивное взаимодействие с тестирующей системой. Это означает, что вы можете делать запросы и получать ответы в online-режиме. Обратите внимание, что ввод/вывод в этой задаче — стандартный (то есть с экрана на экран). После вывода очередного запроса обязательно используйте функции очистки потока, чтобы часть вашего вывода не осталась в каком-нибудь буфере. Например, на C++ надо использовать функцию `fflush(stdout)`, на Java вызов `System.out.flush()` и `stdout.flush()` для языка Python.

В школах Бейтландии собираются ввести новую методику преподавания математики, которая предполагает, что все уроки будут вестись в интерактивно-игровой форме. Одной из самых всеми любимых игр является игра «Дробь», применяемая для изучения рациональных дробей.

Игра предназначена для двух игроков. Вначале игры игрокам предоставляется некоторая несократимая рациональная дробь $\frac{A}{B}$. Далее игроки ходят по очереди начиная с первого игрока. За один ход игрок обязан вычесть единицу либо из числителя дроби, если числитель больше единицы, либо из знаменателя, если знаменатель больше единицы. Если после этого дробь можно сократить, то выполняется сокращение дроби до несократимой.

Далее ход переходит к другому игроку. Игра продолжается до тех пор, пока игрок может сделать ход, то есть выигрывает тот игрок, после хода которого получается дробь $\frac{1}{1} = 1$.

Для примера выше если в начале игры дана дробь $\frac{2}{5}$, то после хода первого игрока может быть получена дробь $\frac{1}{5}$ (вычитанием единицы из числителя) либо дробь $\frac{1}{2}$ (вычитанием единицы из знаменателя и последующего сокращения).

Игра стала настолько популярна, что руководство школы решило в конце учебного года провести чемпионат школы по данной игре. Отличник Петя, учащийся пятого математического класса, решил выиграть этот чемпионат. Петя заметил, что из всех записавшихся на участие в чемпионате он является самым младшим, а также, что все его потенциальные соперники достаточно вежливые люди, которые предоставят Пете право сделать ход первым. Пете сказали, что он сможет победить независимо от ходов второго игрока, если пойдёт первым и будет оптимально ходить. Осознавая свое преимущество, Петя уверен, что можно разработать программу, которая позволит ему для любой заданной дроби одержать победу. Ваша задача — помочь юному дарованию стать чемпионом.

Формат входных данных

В единственной строке входных данных будут даны два целых положительных числа A и B ($1 \leq A, B \leq 2 \cdot 10^5$), задающие начальную дробь. Гарантируется, что всегда существует стратегия, позволяющая Пете выиграть.

Протокол взаимодействия

В каждый ход Пети выведите «A» (без кавычек), если Петя должен уменьшить числитель дроби и «B» (без кавычек), если Петя должен уменьшить знаменатель дроби. **Не забывайте делать операцию «flush»!**

В ответ на ваш запрос программа жюри выведет «A», если второй игрок уменьшил числитель и «B», если второй игрок уменьшил знаменатель.

Как только дробь становится равной 1, ваше решение будет признанно корректным если вы сделали последний ход.

В конце игры программа жюри выведет «OK» если вы выиграли и сделали последний ход и «WA» если вы проиграли и последний ход был сделан программой жюри. После этого ваша программа должна завершить работу.

Обратите внимание на оценку задачи.

Примеры

стандартный ввод	стандартный вывод
3 2 OK	A
5 2 A A OK	B A A

Замечание

В этой задаче есть 30 тестов, каждый оценивается отдельно. В каждом тесте подбираются случайные A и B , такие что они взаимно просты и для них существует выигрышная стратегия. При каждом новом запуске числа A и B новые. У каждого теста есть параметры P_1, P_2, P_3, P_4, P_5 . Программа жюри, которая играет против вас, при возможности уменьшит и числитель и знаменатель с вероятностью P_1 процентов пойдёт случайно, с вероятностью P_2 процентов уменьшит числитель, с вероятностью P_3 процентов уменьшит знаменатель, с вероятностью P_4 процентов пойдёт оптимально и с вероятностью P_5 процентов пойдёт не оптимально.

Обратите внимание, что вашей программе не передаётся номер теста.

№	Ограничения на A	Ограничения на B	P_1	P_2	P_3	P_4	P_5	Балл	Примечания
1	$A = 3$	$B = 2$	0	0	0	100	0	0	тест из условия
2	$A = 5$	$B = 2$	0	0	0	0	100	0	тест из условия
3	$A \leq 20$	$B \leq 20$	1	22	13	64	0	5	—
4	$A \leq 20$	$B \leq 20$	0	0	0	100	0	5	—
5	$A \leq 50$	$B \leq 50$	0	0	0	100	0	5	—
6	$A \leq 1000$	$B \leq 1000$	0	0	0	100	0	5	—
7	$A \leq 1000$	$B \leq 1000$	0	0	0	100	0	5	—
8	$A \leq 1000$	$B \leq 1000$	0	0	0	100	0	5	—
9	$A \leq 1000$	$B \leq 1000$	0	0	0	100	0	5	—
10	$A \leq 1000$	$B \leq 1000$	0	0	0	100	0	5	—
11	$A \leq 10000$	$B \leq 100$	0	0	0	100	0	5	—
12	$A \leq 100$	$B \leq 10000$	0	0	0	100	0	5	—
13	$A \leq 100000$	$B \leq 100000$	11	61	6	18	4	2	—
14	$A \leq 100000$	$B \leq 100000$	100	0	0	0	0	2	—
15	$A \leq 100000$	$B \leq 100000$	0	100	0	0	0	3	—
16	$A \leq 100000$	$B \leq 100000$	0	0	100	0	0	2	—
17	$A \leq 100000$	$B \leq 100000$	0	0	0	100	0	5	—
18	$A \leq 100000$	$B \leq 100000$	0	0	0	0	100	1	—
19	$A \leq 100000$	$B \leq 100000$	26	8	17	12	37	1	—
20	$A \leq 100000$	$B \leq 100000$	100	0	0	0	0	3	—
21	$A \leq 100000$	$B \leq 100000$	0	100	0	0	0	2	—
22	$A \leq 100000$	$B \leq 100000$	0	0	100	0	0	1	—
23	$A \leq 100000$	$B \leq 100000$	0	0	0	100	0	5	—
24	$A \leq 100000$	$B \leq 100000$	0	0	0	0	100	1	—
25	$A \leq 100000$	$B \leq 100000$	3	1	15	37	44	4	—
26	$A \leq 100000$	$B \leq 100000$	0	0	0	100	0	5	—
27	$A \leq 100000$	$B \leq 100000$	43	20	28	9	0	2	—
28	$A \leq 100000$	$B \leq 100000$	0	0	0	100	0	5	—
29	$A \leq 100000$	$B \leq 100000$	0	12	23	10	55	1	—
30	$A \leq 100000$	$B \leq 100000$	0	0	0	100	0	5	—

Задача С. Ханы

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

Элли недавно узнала про Булгарских ханов — правителей кочевых народов, которые путешествовали по континенту сотни лет, прежде чем они наконец поселились навсегда в местах, где сейчас находится Болгария.

Континент, по которому они скитались, разделен на $N * M$ регионов, расположенных в форме прямоугольника с N строками и M столбцами. Хань останавливались ровно на один год в определенном регионе, и пока они жили там, их клан съедал всю еду в этом регионе. В конце года они перемещались в один из (не более чем) четырех соседних по стороне регионов, там они проводили следующий год, съедая всю еду в нем, и так далее. Будем считать, что перемещения в соседний регион происходят мгновенно (в конце концов, что такое несколько дней путешествия по сравнению с целым годом). Хань никогда не проводили два года подряд в одном регионе, в этом случае их клан мог бы погибнуть от голода.

Для каждого региона известно максимальное количество еды, которое может в нем находиться. Обозначим это значение целым числом A_{ij} . После того, как хань съедали всю еду в регионе и уезжали со своим кланом в соседний регион, еда в нем начинала восстанавливаться. Через год после отъезда ханов в регионе появлялась 1 единица еды. После этого каждый год количество еды в регионе удваивалось, пока оно не достигало максимального значения A_{ij} для этого региона. Обратите внимание, что количество еды никогда не превышало максимального количества, которое могло находиться в регионе. Например, если $A_{ij} = 55$, то количество еды в этом регионе в начале каждого из первых десяти лет после отъезда ханов из этого региона, было, соответственно, 0, 1, 2, 4, 8, 16, 32, 55, 55, 55.

Хань никогда не возвращались в регион, пока он не восстанавливался полностью и количество еды в нем не достигало максимума. Из-за этого могла, например, сложиться ситуация, что хань переместились в регион, где меньше еды (скажем, 42 единицы, но это максимальное количество), а не в регион, где больше еды (скажем, 64, а максимум 71). В примере в предыдущем параграфе хань могли бы вернуться в регион в начале 8 года после своего отъезда, это первый год, в который в этом регионе количество еды максимально.

Элли знает информацию о максимальном количестве еды в каждом регионе континента, заданную как матрицу A с N строками и M столбцами. Зная, что хань проведут первый год в левом верхнем регионе, а исходно каждый регион содержит максимальное возможное для этого региона количество еды, выясните, какое максимальное количество еды хань смогут суммарно съесть за K лет.

Формат входных данных

На первой строке входных данных заданы три целых числа N , M , и K ($1 \leq N, M \leq 10$, $1 \leq K \leq 100$), задающих, соответственно, количество строк, количество столбцов в матрице и количество лет. На каждой из следующих N строк находятся по M целых чисел A_{ij} ($10 \leq A_{ij} \leq 100$), задающих максимальное количество еды в соответствующем регионе.

Формат выходных данных

Выведите одну строку, содержащую одно целое число — максимальное суммарное количество еды, которое хань смогут съесть, если они будут путешествовать оптимально.

Гарантируется, что всегда будет путь, который не нарушает правила, что нельзя повторно посещать регион до момента, когда в нем полностью восстановится максимальное количество еды.

Система оценки

В данной задаче каждый тест оценивается отдельно

- В тестах, имеющих стоимость в 20 процентов от баллов за задачу, выполнено $1 \leq N, M \leq 4$

- В тестах, имеющих стоимость в еще 20 процентов от баллов за задачу, выполнено $1 \leq K \leq 20$

Пример

стандартный ввод	стандартный вывод
4 4 11 11 17 13 96 10 12 18 15 13 12 16 17 24 10 14 22	254

Замечание

В первом примере регионы, которые ханы могут посетить, чтобы съесть максимальное количество еды (254 единицы) — регионы с максимальным количеством еды 11, 17, 13, 96, 15, 17, 22, 14, 16, 18, 15, соответственно. При этих перемещениях они посетят дважды лишь один регион — с максимальным количеством еды 15. Обратите внимание, что после последнего года все регионы, соседние с последним регионом, посещенном ханами, не содержат максимального возможного количества еды. Для приведенного теста это не проблема, поскольку это последний год. Но если бы ханам необходимо было продолжить перемещения (например, K было бы равно 12, а не 11), то им пришлось бы выбрать другой путь. Вариант оптимального пути для $K = 12$ по континенту из первого примера: 11, 17, 13, 96, 15, 18, 16, 17, 22, 14, 10, 24, с суммой 273.

Задача D. Лучи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

На плоскости нарисованы N лучей с начальными точками, лежащими на оси Oy . Лучи заданы уравнениями вида $y = A_i \cdot x + B_i$ при $x > 0$ (т. е. все лучи лежат в правой полуплоскости относительно прямой Oy).

Вам надо ответить на Q запросов следующего вида: найти самую правую точку пересечения прямой $y = C_j \cdot x + D_j$ с нарисованными лучами (т. е. точку пересечения с максимальной x -координатой).

Формат входных данных

В первой строке входного файла идёт целое число N — количество лучей ($N \leq 50000$). В последующих N строках идут по два целых числа A_i и B_i ($-10^9 \leq A_i, B_i \leq 10^9$) — коэффициенты i -го луча.

Далее идёт строка, содержащая число Q — количество запросов ($Q \leq 50000$).

В каждой из последующих Q чисел идут по два целых числа E_j и F_j , используемые для генерации коэффициентов для очередного запроса по следующим правилам. Если текущий запрос — первый или текущий запрос не первый и прямая из предыдущего запроса пересекла хотя бы один луч, то очередные $C_j = E_j$, $D_j = F_j$. В противном случае $C_j = E_j \text{ xor } (2^{29} - 1)$, $D_j = F_j \text{ xor } (2^{29} - 1)$, где xor — операция побитового XOR.

Гарантируется, что $-10^9 \leq C_i, D_i \leq 10^9$. Все числа A_i различны, никакое C_j не совпадает с A_i , никакое D_j не совпадает с B_i .

Формат выходных данных

На каждый запрос выведите строку «No cross», если очередная прямая не пересекает ни одного луча, либо максимальную x -координату пересечения. Ваш ответ будет сравниваться с правильным с относительной или абсолютной погрешностью 10^{-6} .

Система оценки

Первый тест — тест из примера. Решение оценивается по группам тестов. Баллы за каждую группу ставятся только при прохождении всей группы целиком.

- Решения, работающие при $N, Q \leq 2000$, оцениваются из 35 баллов.
- Решения, работающие при $N, Q \leq 30000$, $C_j = 0$ для всех j , оцениваются из 15 баллов.
- Решения, работающие при $N, Q \leq 30000$ и условии, что каждая прямая пересекает хотя бы один луч, оцениваются из 15 баллов.
- Решения, работающие при $N, Q \leq 30000$, оцениваются из 25 баллов.
- Решения, работающие при $N, Q \leq 50000$, оцениваются из 10 баллов.

Пример

стандартный ввод	стандартный вывод
2	0.750000000000000000000000
4 2	No cross
-1 0	1.00000000186264514923
3	
-5 3	
0 1	
-5 3	