

## Задача А. Проверка

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

КэВэ — учитель математики в школе. У него есть любимая последовательность чисел  $b_1, b_2, \dots, b_m$ , состоящая из  $m$  **различных** чисел. Когда КэВэ скучно при проверке домашних заданий, он ищет  $b$  в виде подпоследовательности чисел, записанных в тетради ученика.

Как-то раз КэВэ увидел в тетради ученика последовательность из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Учитель решил применить новый метод проверки домашних заданий. А именно, для каждой позиции  $i$  КэВэ ищет минимальный индекс  $r_i$ , такой что в последовательности  $a_i, a_{i+1}, a_{i+2}, \dots, a_{r_i}$  можно встретить  $b$  как подпоследовательность. Помогите ему для каждого  $i$  найти нужное  $r_i$  или укажите, что такого не найдётся.

Напомним, что подпоследовательностью для  $a_1, a_2, \dots, a_n$  называется набор элементов  $a$ , полученный из  $a_1, a_2, \dots, a_n$  удалением некоторых ее элементов без изменения порядка следования оставшихся.

### Формат входных данных

В первой строке вводится два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 200\,000$ ) — длина последовательности  $a$  и  $b$  соответственно.

Во второй строке вводится  $n$  целых чисел  $a_1, a_2, a_3, \dots, a_n$  ( $1 \leq a_i \leq 200\,000$ ).

В третьей строке вводится  $m$  целых чисел  $b_1, b_2, b_3, \dots, b_m$  ( $1 \leq b_i \leq 200\,000$ ). Гарантируется, что все числа последовательности  $b_i$  различны.

### Формат выходных данных

В единственной строке выведите  $n$  чисел,  $i$ -е из них должно быть равно искомому индексу  $r_i$ . В случае, если такого  $r_i$  не найдётся, выведите вместо него число  $-1$ .

### Примеры

стандартный ввод	стандартный вывод
7 3 1 2 1 3 1 2 1 1 3 2	6 6 6 -1 -1 -1 -1
10 2 1 2 3 4 5 1 2 3 4 5 3 4	4 4 4 9 9 9 9 9 -1 -1

### Замечание

В первом примере  $b$  обязательно закончится в позиции 6, потому что это единственное число 2, стоящее после числа 3. Например, элементы с позиций  $\{3, 4, 6\}$  дают нужную подпоследовательность

Во втором примере подпоследовательность  $b$  встречается в элементах с позиций  $\{3, 4\}$  и  $\{8, 9\}$ .

### Система оценки

Тесты к этой задаче состоят из шести групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$m$		
0	0	–	–	–	Тесты из условия.
1	10	$n \leq 50$	$m \leq 50$	0	
2	11	$n \leq 1000$	$m \leq 1000$	0, 1	
3	12	$n \leq 10000$	$m \leq 10000$	0, 1, 2	
4	22	–	$m \leq 2$		
5	15	–	$m \leq 1000$	0, 1, 2, 4	
6	30	–	–	0, 1, 2, 3, 4, 5	

## Задача В. Ремонт дороги

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Наступила осень, стали лить дожди, а коммунальные службы Берляндии начали ремонт дорог. В результате многочисленных дорожных реформ Берляндии, в ней осталась только одна дорога, которая состоит из участков, последовательно пронумерованных целыми числами от  $-10^9$  до  $10^9$ . Решено было ремонтировать некоторые участки главной дороги Берляндии на отрезке с  $l$ -го по  $r$ -й. К сожалению, иногда некоторые участки дорог затапливаются дождями, поэтому ремонт таких участков дорог невозможен.

Изначально уровень воды на каждом участке дороги равен 0. Далее происходят  $n$  событий:

1. Ремонтные службы хотят узнать, можно ли ремонтировать  $x$ -й участок дороги. Для этого им нужно узнать текущий уровень воды на  $x$ -м участке дороги.
2. Проходит ливень над  $x$ -м участком дороги, в результате чего уровень воды на нём увеличивается на 1.

Если после этого где-то уровень воды поднялся до 2, то вода начинает перетекать. На каждом участке дороги с номером  $i$ , где уровень воды был 2, он опускается до 0, а на участках с номерами  $i - 1$  и  $i + 1$  уровень воды увеличивается на 1. Все такие перетекания происходят одновременно. Если после этого на каких-то участках уровень воды снова поднялся до 2, то процесс повторяется одновременно для них всех и продолжается до тех пор, пока на всех участках уровень воды не будет меньше 2. Можно показать, что такой процесс завершится, а также что никогда промежуточный уровень воды не поднимется выше 2. Следующее событие произойдёт только после окончания всех перетеканий. Также гарантируется, что вода никогда не вытечет за пределы участков дороги с  $l$ -го по  $r$ -й.

Вам необходимо ответить на все запросы первого типа.

### Формат входных данных

Первая строка входных данных содержит три целых числа  $n$ ,  $l$  и  $r$  ( $1 \leq n \leq 200\,000$ ,  $-10^9 \leq l \leq r \leq 10^9$ ) — количество запросов и ограничения на номера участков из запросов.

В следующих  $n$  строках вводятся по символу  $c_i$  и целому числу  $x_i$  ( $l \leq x_i \leq r$ ).

- Если  $c_i$  равняется «?», то в  $i$ -м запросе требуется определить уровень воды на  $x_i$ -м участке дороги.
- Если  $c_i$  равняется «+», то в  $i$ -м запросе уровень воды на  $x_i$ -м участке увеличивается на единицу.

Гарантируется, что вода никогда не вытечет за пределы отрезка  $[l, r]$ .

### Формат выходных данных

Для каждого запроса первого типа в отдельной строке выведите одно целое число (0 или 1) — уровень воды на участке из запроса.

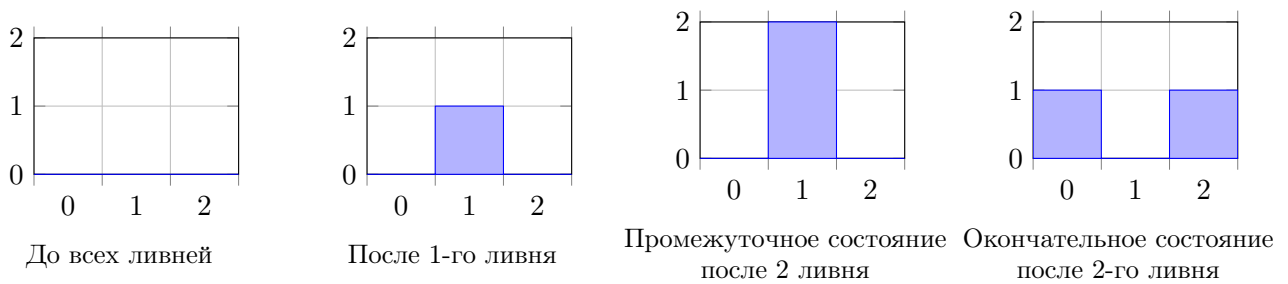
## Примеры

стандартный ввод	стандартный вывод
5 0 2 + 1 + 1 ? 0 ? 1 ? 2	1 0 1
7 0 4 + 1 + 2 + 3 + 2 ? 0 ? 2 ? 4	1 0 1
10 -5 5 + 0 + -1 + 1 ? -2 ? 0 + -1 ? -1 ? 0 ? 1 ? 2	0 1 1 1 0 1

## Замечание

Ниже представлены картинки, поясняющие первые два теста из условия.

По вертикальной оси отмечены уровни воды, а по горизонтальной — координаты.



**Рис. 1: Первый тест из условия**

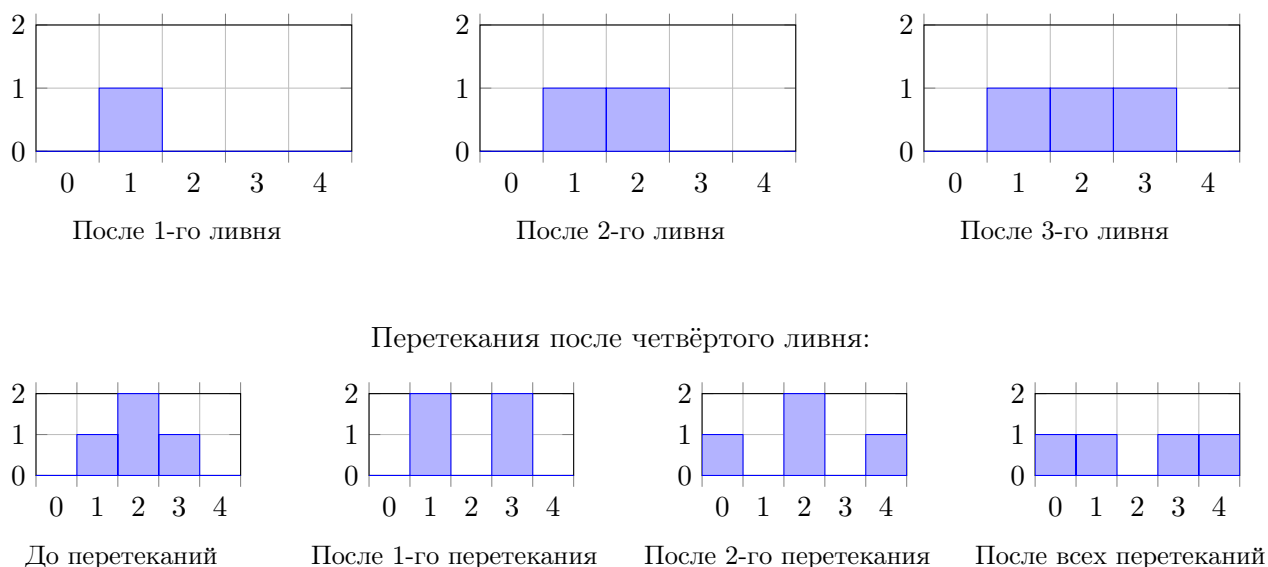


Рис. 2: Второй тест из условия

## Система оценки

Тесты к этой задаче состоят из 4 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		$n$	$l$	$r$		
0	0	–	–	–	–	Тесты из условия.
1	26	$n \leq 100$	$l = 0$	$r = 100$	–	
2	21	$n \leq 1000$	$l = -10^9$	$r = 10^9$	0, 1	
3	22	$n \leq 10\,000$	$l = 0$	$r = 1000$	1	
4	31	–	$l = -10^9$	$r = 10^9$	0, 1, 2, 3	<b>Offline-проверка.</b>

## Задача С. Новые кампусы!

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	5 секунд
Ограничение по памяти:	1024 мегабайта

Недавно вы стали ректором одного из университетов и решили открыть в нем новую программу! На ней вы будете учить студентов спортивному программированию. Поэтому у них будут два типа занятий: спорт (чтобы развить силу рук) и программирование. Основным достоинством этой программы будет обучение в двух кампусах одновременно — по четным дням студенты будут ездить в первый кампус, а по нечетным — во второй.

Оба кампуса вашего университета устроены очень необычно: в каждом из них есть по  $n$  аудиторий, пронумерованных от 1 до  $n$ , и по  $n - 1$  переходу между ними, при этом из любой аудитории можно добраться в любую другую по переходам.

Однако вы обнаружили, что студентам сложно ориентироваться сразу в двух кампусах, и решили упростить им жизнь. Вы решили выбрать два номера аудиторий  $u$  и  $v$  ( $u \neq v$ ): в аудитории с номером  $u$  студенты будут заниматься спортом, а в аудитории с номером  $v$  — программированием. Обратите внимание, что  $u$  и  $v$  выбираются одинаковыми для обоих кампусов.

Так как вы хотите, чтобы студенты тратили меньше времени на перемещение между аудиториями, вам нужно минимизировать суммарное расстояние, которое потребуется преодолеть студентам между выбранными аудиториями в каждом из кампусов. Более формально, вам нужно найти такие номера  $u, v$ , что  $d_1(u, v) + d_2(u, v)$  минимально, где  $d_1(u, v)$  — это расстояние между аудиториями  $u$  и  $v$  в первом кампусе, а  $d_2(u, v)$  — во втором. Расстоянием между аудиториями называется минимальное число переходов, через которые нужно пройти, чтобы добраться из одной аудитории в другую.

В обоих кампусах есть вход, и он ведет в аудиторию 1. Для всех остальных аудиторий разработан план эвакуации. В первом кампусе для  $i$ -й аудитории  $p_i$  равно номеру следующей аудитории на пути из  $i$ -й аудитории в первую. Во втором кампусе для  $i$ -й аудитории  $q_i$  равно номеру следующей аудитории на пути из  $i$ -й аудитории в первую.

### Формат входных данных

В первой строке входных данных содержится одно целое число  $n$  ( $2 \leq n \leq 10^6$ ) — количество аудиторий.

В следующей строке находятся  $n - 1$  целых чисел  $p_2, p_3, p_4, \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  — это следующая (кроме  $i$ ) аудитория на пути от  $i$ -й до первой в первом кампусе.

В следующей строке находятся  $n - 1$  целых чисел  $q_2, q_3, q_4, \dots, q_n$  ( $1 \leq q_i \leq n$ ), где  $q_i$  — это следующая (кроме  $i$ ) аудитория на пути от  $i$ -й до первой во втором кампусе.

### Формат выходных данных

В первой строке выведите минимальную величину  $d_1(u, v) + d_2(u, v)$ .

Во второй строке выведите любую пару вершин  $u, v$ , таких что  $d_1(u, v) + d_2(u, v)$  минимально.

## Примеры

стандартный ввод	стандартный вывод
5 3 1 2 3 5 4 1 3	2 5 3
5 5 1 2 3 4 4 1 4	2 2 4
7 1 2 2 7 1 3 5 5 5 1 5 2	3 2 1
9 5 2 1 4 9 8 3 7 1 4 7 9 8 2 5 3	4 2 1

## Замечание

В первом примере в первом кампусе есть переходы между аудиториями 3 и 2, 1 и 3, 2 и 4, 3 и 5. Во втором кампусе есть переходы между аудиториями 5 и 2, 4 и 3, 1 и 4, 3 и 5.

## Система оценки

Тесты к этой задаче состоят из 12 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Длинный тур отборочного этапа Открытой олимпиады школьников 2021–2022 учебного года  
Россия. 1 ноября 2021 – 23 января 2022

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		$n$		
0	0	–	–	Тесты из условия.
1	12	$n \leq 500$	0	
2	11	$n \leq 5000$	0, 1	
3	8	$n \leq 50\,000$	0, 1, 2	
4	11	$n \leq 100\,000$	–	В первом кампусе существует аудитория, соединенная прямыми переходами со всеми остальными аудиториями
5	12	$n \leq 100\,000$	–	В обоих кампусах для каждой аудитории существует не более двух переходов в соседние аудитории
6	10	$n \leq 100\,000$	5	В первом кампусе для каждой аудитории существует не более двух переходов в соседние аудитории
7	9	$n \leq 100\,000$	0 – 6	
8	10	$n \leq 200\,000$	0 – 7	
9	11	$n \leq 300\,000$	0 – 8	<b>Offline-проверка.</b>
10	3	$n \leq 500\,000$	0 – 9	<b>Offline-проверка.</b>
11	2	$n \leq 750\,000$	0 – 10	<b>Offline-проверка.</b>
12	1	–	0 – 11	<b>Offline-проверка.</b>



## Задача D. Древо жизни

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Представим, что судьба каждого человека predetermined: существует фиксированное корневое дерево из  $n$  вершин, которое называется деревом жизни. Напомним, что деревом называется связный граф с  $n$  вершинами и  $n - 1$  ребром. В корневом дереве есть выделенная вершина, называемая корнем. Предком вершины  $i$  называется ближайшая к  $i$  вершина на пути от корня к  $i$ .

Всего есть  $n$  возможных событий, счастье от наступления  $i$ -го события равно  $a_i$ . Событие называется радостным, если  $a_i \geq 0$ . Рассмотрим произвольную расстановку событий в дереве жизни, где каждой вершине соответствует ровно одно событие, каждое событие соответствует какой-либо вершине. Справедливостью судьбы назовем суммарное счастье событий, достижимых в дереве из корня, если разрешено перемещаться только по радостным событиям. Иначе говоря, таких событий, что существует путь из корня в вершину с этим событием, проходящий только по радостным событиям. Требуется просуммировать справедливость судьб по всем возможным способам расставить события в вершинах дерева. Две расстановки считаются различными, если существует вершина дерева, которой соответствуют события с разными номерами.

Так как ответ может быть очень большим, посчитайте его по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке входных данных содержится целое число  $n$  ( $2 \leq n \leq 200\,000$ ) — количество вершин в дереве судьбы и количество событий.

В следующей строке содержится  $n - 1$  целое число  $p_2, p_3, p_4, \dots, p_n$  ( $1 \leq p_i < i$ ), где  $p_i$  — предок  $i$ -й вершины в дереве судьбы. Корнем дерева судьбы является вершина номер 1.

В следующей строке содержится  $n$  целых чисел  $a_i$  ( $|a_i| \leq 10^9$ ) — счастье от наступления каждого события.

### Формат выходных данных

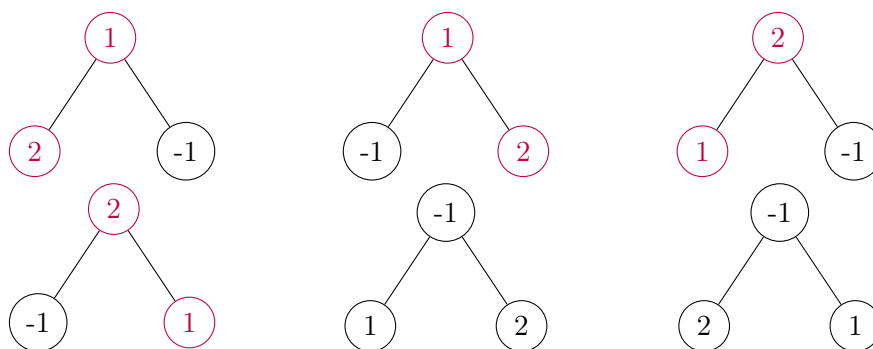
Выведите единственное целое число — искомую сумму по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
3 1 1 1 2 -1	12
4 1 2 1 2 0 3 1	144
7 1 1 2 1 2 3 4 -1 2 0 -3 9 -1	33480

### Замечание

Ниже представлены 6 деревьев, соответствующих всем возможным расстановкам событий в первом тесте из условия, цветом выделены достижимые радостные события.



## Система оценки

Тесты к этой задаче состоят из 7 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		$n$		
0	0	–	–	Тесты из условия.
1	15	$n \leq 10$	0	
2	10	$n \leq 20$	0, 1	
3	12	–	–	$p_i = 1$
4	16	$n \leq 200$	0, 1, 2	
5	13	$n \leq 2000$	–	$p_i = i - 1$
6	15	–	5	$p_i = i - 1$
7	19	–	0–6	

## Задача Е. Битовый хаос

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 1024 мегабайта

У вас есть массив чисел  $a$  длины  $n$ . Вам нужно научиться обрабатывать следующие запросы:

- `upd and l r x` — сделать  $a_i = a_i \& x$  (битовая операция И) для всех  $l \leq i \leq r$ .
- `upd or l r x` — сделать  $a_i = a_i | x$  (битовая операция ИЛИ) для всех  $l \leq i \leq r$ .
- `upd xor l r x` — сделать  $a_i = a_i \oplus x$  (битовая операция XOR) для всех  $l \leq i \leq r$ .
- `get and l r` — найти побитовое И всех  $a_i$ , где  $l \leq i \leq r$ .
- `get or l r` — найти побитовое ИЛИ всех  $a_i$ , где  $l \leq i \leq r$ .
- `get xor l r` — найти побитовый XOR всех  $a_i$ , где  $l \leq i \leq r$ .

### Формат входных данных

В первой строке вводятся два целых числа  $n, q$  ( $1 \leq n, q \leq 500\,000$ ) — размер массива  $a$  и количество запросов.

Во второй строке вводится  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^{18}$ ).

В следующих  $q$  строках будет информация о запросах по одному на строке согласно условию задачи ( $1 \leq l \leq r \leq n, 0 \leq x \leq 10^{18}$ ).

### Формат выходных данных

На каждый запрос типа `get` выведите найденный ответ. Гарантируется, что в каждом тесте есть хотя бы один такой запрос.

### Пример

стандартный ввод	стандартный вывод
10 10	15
5 7 3 10 12 0 11 15 14 31	8
get or 3 7	19
get xor 2 9	8
upd and 1 5 14	19
upd xor 3 10 7	15
get xor 1 10	7
get and 9 10	
upd or 3 8 6	
get xor 1 10	
get or 5 5	
get xor 4 6	

### Система оценки

Тесты к этой задаче состоят из девяти групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых групп.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$q$		
0	0	–	–	–	Тесты из условия.
1	11	$n \leq 1000$	$q \leq 1000$	0	
2	11	–	–		Применяются только операции <i>upd xor</i> и <i>get xor</i>
3	13	–	–		Нет запросов изменения
4	12	–	–		Применяются только операции <i>upd and</i> и <i>get xor</i>
5	12	–	$q \leq 50\,000$	0	Значения в массиве и в запросах изменения не превосходят $10^9$ .
6	13	–	–		Применяются только операции <i>upd xor</i> и <i>get and</i>
7	12	–	–		Применяются операции, связанные с <i>and</i> и <i>or</i>
8	12	$n \leq 300\,000$	$q \leq 300\,000$	0, 1	
9	4	–	–	0 – 8	

## Задача F. Финальная Битва

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Маленький Тайлер почти прошёл свою любимую видеоигру, ему осталось только сражение с главным боссом. За время игры он собрал целую кучу различных героев, до начала боя они упорядочены в боевой строй — таблицу с  $n$  строками и  $m$  столбцами. Урон персонажа в  $i$ -й строке и  $j$ -м столбце равен  $a_{ij}$ .

Перед началом битвы он должен принести в жертву ровно  $k$  из своих персонажей. После того как все жертвы принесены, на месте каждой из них появится Голем с уроном, равным сумме уронов оставшихся персонажей в соответствующих строке и столбце. После этого все персонажи на поле, включая Големов, нанесут по одному удару.

Тайлер ещё не знает, насколько силен главный босс, поэтому ему интересно, какой максимальный суммарный урон он сможет нанести. Помогите ему это посчитать.

### Формат входных данных

В первой строке вводятся три целых числа  $n, m$  и  $k$  ( $1 \leq n, m \leq 300, 1 \leq k \leq \min(6, nm)$ ) — размеры таблицы и количество жертв, которое необходимо принести.

В каждой из следующих  $n$  строк вводятся  $m$  целых чисел. В  $i$ -й из этих строк на позиции  $j$  вводится число  $a_{ij}$  ( $0 \leq a_{ij} \leq 2 \cdot 10^5$ ) — урон соответствующего персонажа.

### Формат выходных данных

Выведите единственное число — максимальный суммарный урон, который смогут нанести персонажи, если оптимальным образом выбрать, кого принести в жертву.

### Примеры

стандартный ввод	стандартный вывод
2 3 1 0 6 2 2 3 6	30
2 4 2 4 4 1 1 2 2 9 9	72
2 2 3 12 6 3 9	36

### Замечание

Пояснения к тесту 1:

Если принести в жертву персонажа на позиции  $(1, 1)$ , то искомая сумма будет 29.

0	6	2
2	3	6

10	6	2
2	3	6

Если принести в жертву персонажа на позиции  $(2, 2)$ , то искомая сумма будет 30.

0	6	2
2	3	6

0	6	2
2	14	6

Пояснения к тесту 2:

Если принести в жертву персонажей на позициях  $(2, 1)$  и  $(2, 2)$ , то на каждой из этих позиций появится Голем с силой  $4 + 9 + 9 = 22$ , а сумма во все таблице станет равной 72.

4	4	1	1
2	2	9	9

4	4	1	1
22	22	9	9

## Система оценки

Тесты к этой задаче состоят из 7 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n, m$	$k$		
0	0	–	–	–	Тесты из условия.
1	8	–	$k \leq 1$		
2	11	–	$k \leq 2$	1	
3	19	–	$k \leq 3$	0, 1, 2	
4	14	$n, m \leq 100$	$k \leq 4$	0	
5	9	–	$k \leq 4$	0 – 4	
6	17	–	$k \leq 5$	0 – 5	
7	22	–	–	0 – 6	<b>Offline-проверка.</b>

## Задача G. Гармоничные шарфы

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Наступила зима, и вы решили, что пора обзавестись шарфом. У вас есть полосатый кусок ткани,  $i$ -я полоска имеет цвет  $s_i$ . Для удобства цвета обозначены строчными буквами латинского алфавита. Последовательность полосок называется гармоничной и подходит для шарфа, если в ней встречаются ровно два различных цвета. Например, последовательности "ad", "bcbbc" являются гармоничными, а "d", "aaa", "acb" не являются.

Одного шарфа вам может не хватить, поэтому вы заинтересованы в полезных кусках ткани — непустых кусках, которые можно разделить на несколько непрерывных последовательностей полосок, являющихся гармоничными. Например, последовательности "ab", "cbcac" являются полезными, так как их можно разделить на куски "ab" и "cbc" + "ac" соответственно.

Посчитайте количество способов вырезать из вашей ткани непрерывный полезный кусок.

### Формат входных данных

В первой строке входных данных вводится одно целое число  $n$  ( $1 \leq n \leq 1\,000\,000$ ) — количество полосок в вашем куске ткани.

Во второй строке входных данных вводится строка  $s$  длины  $n$  — последовательность цветов полосок в вашей ткани.

### Формат выходных данных

Выведите одно целое число — количество полезных кусков в вашей ткани.

### Примеры

стандартный ввод	стандартный вывод
4 baba	6
4 cbca	5

### Замечание

Во первом тесте из условия полезными являются куски "ba", "ba", "ab", "bab", "aba", "baba". Обратите внимание, что различные полезные куски могут совпадать как строки.

Во втором тесте из условия полезными являются куски "cb", "cbc", "bc", "ca", "cbca".

### Система оценки

Тесты к этой задаче состоят из 5 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов необходимых групп.

Группа	Баллы	Доп. ограничения	Необх. группы	Комментарий
		$n$		
0	0	–	–	Тесты из условия.
1	17	$n \leq 500$	0	
2	14	$n \leq 5000$	0	$s_i \neq s_{i-1}$
3	19	$n \leq 5000$	0, 1, 2	
4	21	–	0, 2	$s_i \neq s_{i-1}$
5	29	–	0 – 4	

## Задача Н. Заработок без вложений

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Жаба Клава узнала про инновационную криптовалюту MMMcoin и решила, что это отличная возможность заработать.

Аккаунт в MMMcoin представляет собой цепочку из  $k$  кошельков, пронумерованных от 1 до  $k$ . Обозначим количество монеток на счете  $i$ -го кошелька за  $a_i$ . Изначально все  $a_i$  равны 0.

Система MMMcoin поддерживает два типа операций с кошельками:

- Увеличить сумму на счете на единицу. Иначе говоря, присвоить  $a_i = a_i + 1$  для выбранного  $i$ .
- Прибавить к сумме на счете количество денег на счетах с меньшими номерами. Иначе говоря, присвоить  $a_i = a_1 + a_2 + \dots + a_i$  для выбранного  $i$ .

Клава рассматривает  $t$  независимых сценариев обогащения, в  $i$ -м из них она хочет получить  $n_i$  монеток на счете любого из кошельков, используя не более  $q$  операций. Можно показать, что для заданных ограничений это всегда возможно. Помогите ей!

### Формат входных данных

В первой строке входных данных вводятся три целых числа  $k, q, t$  ( $1 \leq k \leq 100, 1 \leq q \leq 100, 1 \leq t \leq 5000$ ) — количество кошельков, ограничение на число операций и количество сценариев соответственно.

В каждой из следующих  $t$  строк вводится одно целое число  $n_i$  ( $1 \leq n_i \leq 1\,000\,000$ ). Гарантируется, что все  $n_i$  различны.

### Формат выходных данных

Для каждого из  $t$  сценариев выведите число  $k_i$  — количество использованных операций. В следующих  $k_i$  строках выведите операции в следующем формате:

- 0  $i$  ( $1 \leq i \leq k$ ) - для операций прибавления 1 к сумме на счете с номером  $i$ .
- 1  $i$  ( $1 \leq i \leq k$ ) - для операций прибавления к сумме на счете  $i$  суммы на счетах с меньшими номерами.

### Пример

стандартный ввод	стандартный вывод
100 100 3	1
1	0 1
2	3
3	0 1
	1 2
	1 2
	4
	0 1
	0 3
	0 2
	1 3

### Система оценки

Тесты к этой задаче состоят из 3 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп



Группа	Баллы	Доп. ограничения				Необх. группы	Комментарий
		$k$	$q$	$t$	$n_i$		
0	0	–	–	–	–	–	Тесты из условия.
1	25	$k = 100$	$q = 100$	$t = 100$	$n_i \leq 100$	0	
2	32	$k = 11$	$q = 40$	$t = 1000$	$n_i \leq 1000$	0, 1	
3	43	$k = 10$	$q = 40$	$t = 5000$	$n_i \leq 1\,000\,000$	0, 1, 2	

## Задача I. Прибытие

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Радиотелескоп РАТАН-600 засек НЛО на подлете к Земле. НЛО представляет собой две касающиеся окружности неизвестного радиуса  $r$ . По предварительным подсчетам он совершит посадку на участке сибирской тайги. Участок тайги представляет собой прямоугольник размером  $x$  на  $y$ , на котором растут  $n$  ёлок,  $i$ -я ёлка расположена в точке с координатами  $x_i, y_i$ .

НЛО будет посажен так, что все точки на его границе окажутся строго внутри ограничивающего прямоугольника. Ученых интересует максимальный радиус НЛО  $r$  такой, что можно посадить корабль, не повредив ни одной ёлки. Ёлка считается поврежденной, если она оказалась строго внутри одной из окружностей, составляющих корабль.

### Формат входных данных

В первой строке вводится  $t$  ( $1 \leq t \leq 1000$ ) — количество наборов входных данных. В следующих строках содержатся описания каждого тестового набора.

В первой строке каждого описания вводятся два целых числа  $x, y$  ( $2 \leq x, y \leq 10\,000$ ) — размеры ограничивающего прямоугольника.

Во второй строке вводится одно целое число  $n$  ( $1 \leq n \leq 1000$ ) — количество ёлок.

В следующих  $n$  строках вводятся по два целых числа  $x_i, y_i$  ( $1 \leq x_i \leq x - 1, 1 \leq y_i \leq y - 1$ ) — координаты ёлок.

Гарантируется, что координаты всех ёлок различны в каждом из наборов входных данных.

### Формат выходных данных

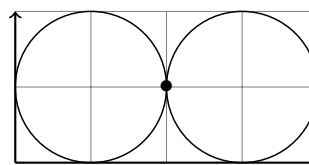
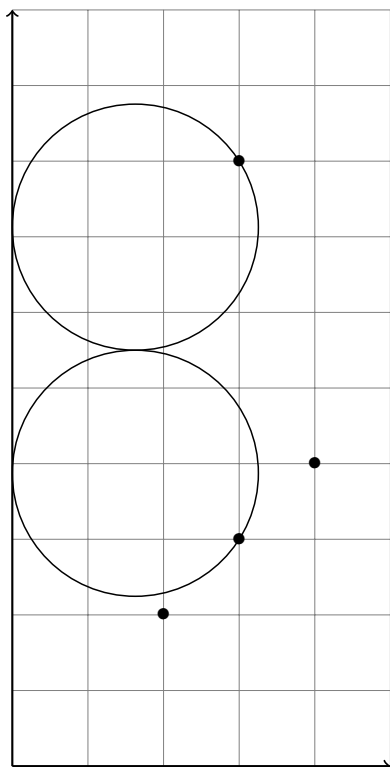
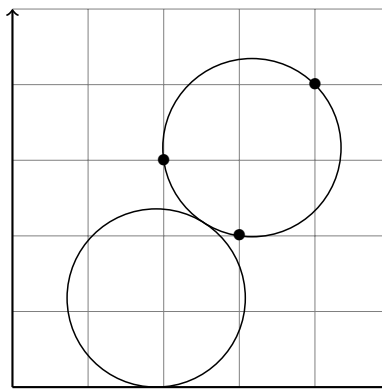
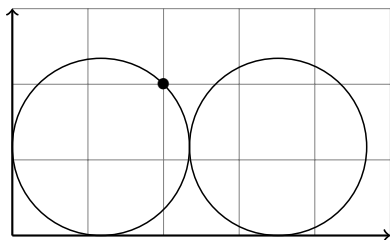
Для каждого тестового набора выведите вещественное число  $r$  — максимальный радиус НЛО с абсолютной или относительной погрешностью не более  $10^{-6}$ .

## Примеры

стандартный ввод	стандартный вывод
4 5 3 1 2 2 5 5 3 3 2 2 3 4 4 5 10 4 3 8 2 2 3 3 4 4 4 2 1 2 1	1.1715729 1.1785113 1.6270167 1.0000000
1 10 10 10 1 4 2 7 3 2 5 8 6 1 6 2 6 7 7 2 7 8 8 5	1.9649222
1 1505 1671 10 38 912 147 740 149 266 259 459 360 802 513 523 692 299 717 1185 1114 362 1363 1536	368.1802590

## Замечание

Расположения окружностей, соответствующих ответам на первый тест из условия.



## Система оценки

Тесты к этой задаче состоят из 9 групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов некоторых из предыдущих групп. Обратите внимание, прохождение тестов из условия не требуется для некоторых групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обозначим за  $N$  сумму  $n_i$  по всем тестовым набором внутри одного теста.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n_i$	$N$		
0	0	–	–	–	Тесты из условия.
1	4	$n_i = 1$	$N \leq 10$	–	
2	7	$n_i \leq 2$	$N \leq 10$	1	
3	18	$n_i \leq 10$	$N \leq 10$	0 – 2	
4	13	$n_i \leq 30$	$N \leq 30$	0 – 3	
5	10	$n_i \leq 50$	$N \leq 50$	0 – 4	
6	6	$n_i \leq 100$	$N \leq 100$	0 – 5	
7	8	$n_i \leq 300$	$N \leq 300$	0 – 6	
8	14	$n_i \leq 500$	$N \leq 500$	0 – 7	
9	20	$n_i \leq 1000$	$N \leq 1000$	0 – 8	<b>Offline-проверка.</b>