

## Общая информация по задачам второго тура

Задача	Тип задачи	Ограничения
5. Качественный отдых	стандартная	1 с, 1024 МБ
6. Лягушки на болоте	стандартная	1 с, 128 МБ
7. Минимизация инверсий	стандартная	1 с, 1024 МБ
8. Жизнь программистов	стандартная	2 с, 1024 МБ

Необходимо считывать данные из стандартного потока ввода. Выходные данные необходимо выводить в стандартный поток вывода.

Баллы за подзадачу начисляются только, если все тесты этой и необходимых подзадач пройдены. Решение запускается на тестах для определенной подзадачи, если все тесты всех необходимых подзадач пройдены.

Во всех подзадачах каждой задачи во время тура вам показываются баллы за подзадачу, если все тесты пройдены, либо первая ошибка и номер теста.

Для некоторых подзадач может также требоваться, чтобы были пройдены все тесты из условия. Для таких подзадач указана дополнительно буква У.

## Задача 5. Качественный отдых

Ограничение по времени: 1 секунда  
Ограничение по памяти: 1024 мегабайта

Проход проходит стажировку продолжительностью  $n$  календарных дней в ИТ-компании. Проход стажировается в службе поддержки, поэтому у него сложный график рабочих и выходных дней на время стажировки.

Кроме выходных, у Прохода есть некоторое количество отгулов — дополнительных выходных дней, которые он может взять в любые рабочие дни.

За один выходной день Проход качественно отдохнуть не сможет, поэтому он считает *днями качественного отдыха* только те выходные дни, которые входят в последовательность из идущих подряд двух или более выходных дней.

Вам даны  $q$  запросов — различных значений количества отгулов, которые может взять Проход. Ваша задача — по заданному графику рабочих и выходных дней стажировки определить для каждого запроса, какое максимальное количество дней качественного отдыха за время стажировки может получить Проход.



### Формат входных данных

Первая строка входных данных содержит два целых числа  $n$  и  $q$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq q \leq n + 1$ ).

Следующая строка содержит строку  $s$  длины  $n$ , состоящую из символов «0» и «1» — график стажировки. В этой строке символом «0» обозначается рабочий день, а символом «1» — выходной.

В следующих  $q$  строках находятся  $q$  целых чисел  $k_i$  ( $0 \leq k_i \leq n$ ) — количество отгулов в  $i$ -м запросе. Гарантируется, что каждое значение  $k_i$  не превосходит количества рабочих дней в графике стажировки.

### Формат выходных данных

Выведите  $q$  целых чисел — для каждого значения  $k_i$  определите наибольшее количество качественных дней отдыха, которое может получить Проход за время стажировки, выбрав  $k_i$  дополнительных выходных дней.

### Система оценивания

Подзадача	Баллы	Дополнительные ограничения			Необх. подзадачи
		$n$	$q$	дополнительно	
1	6	–	–	Все дни графика — рабочие	–
2	11	–	–	Выходные и рабочие дни чередуются, первый день стажировки — выходной	–
3	12	–	$q = 1$	$k_1 = 0$	–
4	19	–	$q = 1$	$k_1 = 1$	–
5	11	$n \leq 15$	–	–	У
6	17	$n \leq 1000$	–	–	У, 5
7	13	–	–	В графике нет двух выходных подряд	1, 2
8	11	–	–	–	У, 1–7

## Примеры

стандартный ввод	стандартный вывод
3 4 000 0 1 2 3	0 0 2 3
4 3 1010 0 1 2	0 3 4
11 6 110101001 5 2 0 1 4 3	11 7 2 5 10 9

## Замечание

В первом примере все три дня стажировки являются рабочими. Если взять менее двух отгулов, дней качественного отдыха получить невозможно. Для  $k_3 = 2$  или  $k_4 = 3$  можно выбрать отгулами первые  $k_j$  дней стажировки, и все они будут днями качественного отдыха.

Во втором примере один отгул выгодно взять во второй день стажировки, тогда первые три дня стажировки будут днями качественного отдыха.

## Задача 6. лягушки на болоте

Ограничение по времени: 1 секунда  
Ограничение по памяти: 128 мегабайт

В Сочи при подготовке Олимпиады-2014 была завезена самшитовая огнёвка (небольшая бабочка с Дальнего Востока). Она уничтожила самшитовую рощу, поэтому древесным лягушкам теперь приходится жить на болоте. Но они сохранили способность после прыжка менять свой цвет с зелёного на коричневый и наоборот.



Болото представляет собой плоскость, в некоторых точках которой располагаются кочки. Размером кочек можно пренебречь и считать их точками на плоскости. За один прыжок лягушка может перепрыгнуть с кочки, на которой она находится, на любую другую кочку, которая находится от неё на расстоянии не более  $r$ . После каждого прыжка цвет лягушки меняется на противоположный. Прыгать на месте лягушка не умеет.

Вам необходимо для каждой стартовой кочки лягушки от 1 до  $n$  определить, может ли она, совершив некоторое количество прыжков, вернуться на стартовую кочку, поменяв при этом свой цвет.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $r$  ( $2 \leq n \leq 10^5$ ,  $1 \leq r \leq 10^9$ ) — число кочек на болоте и расстояние, на которое прыгает лягушка.

Каждая из следующих  $n$  строк описывает расположение кочек. В  $i$ -й из них содержатся два целых числа  $x_i$  и  $y_i$  ( $0 \leq x_i, y_i \leq 5 \cdot 10^8$ ) — координаты  $i$ -й кочки.

Никакие две кочки не располагаются в одной точке.

### Формат выходных данных

Выведите строку, состоящую из  $n$  символов. Если лягушка, стартовав с кочки  $i$ , может вернуться на неё, имея противоположный цвет,  $i$ -й символ должен быть «1», а иначе — «0».

### Система оценивания

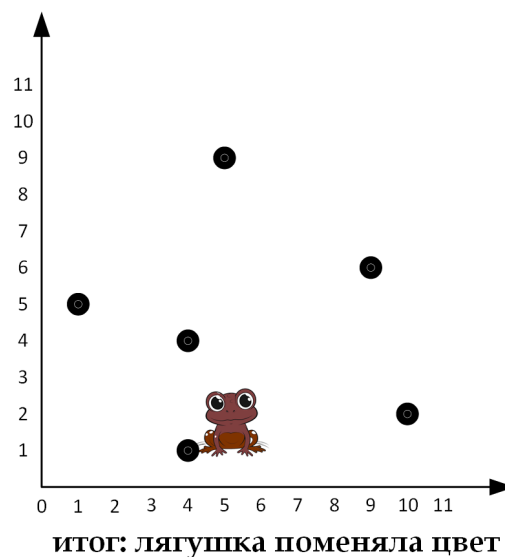
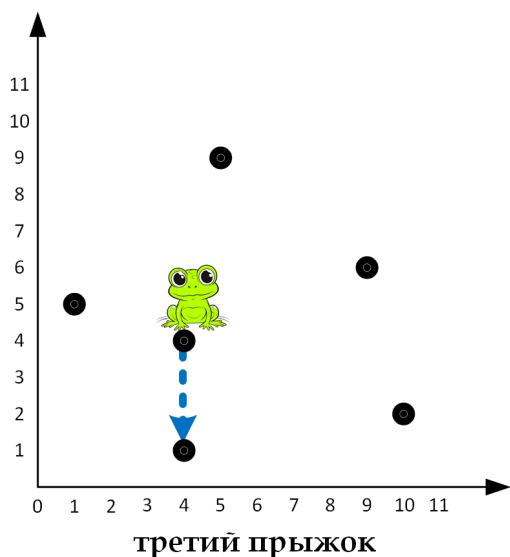
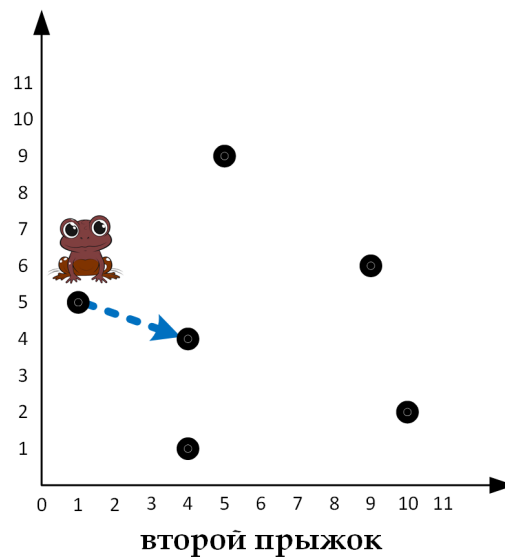
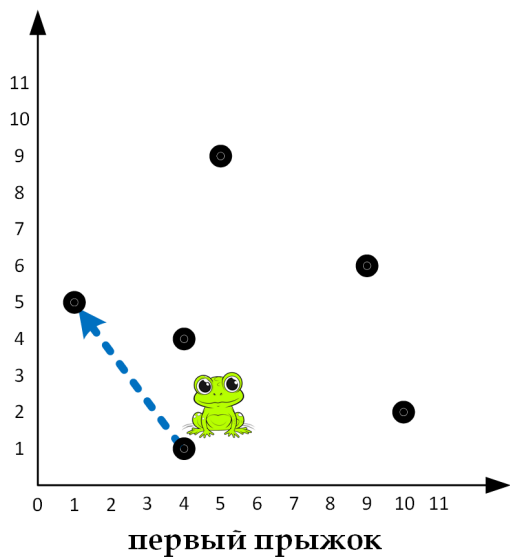
Подзадача	Баллы	Дополнительные ограничения	Необх. подзадачи
1	10	$n \leq 3$	
2	20	$n \leq 200$	У, 1
3	6	$n \leq 1\,000$	У, 1, 2
4	9	$n \leq 10\,000$	У, 1–3
5	16	$y_i = 0$	
6	5	$r \leq 2$	
7	5	$r \leq 4$	6
8	5	$r \leq 10$	У, 6, 7
9	12	$(x_i - x_j)^2 + (y_i - y_j)^2 \geq \frac{r^2}{4}$ , $i \neq j$	У, 6
10	12		У, 1–9

## Пример

стандартный ввод	стандартный вывод
6 5 4 1 4 4 1 5 5 9 9 6 10 2	111000

## Замечание

Прыжки, которые позволяют лягушке поменять цвет, начав с первой кочки, показаны на рисунке ниже.

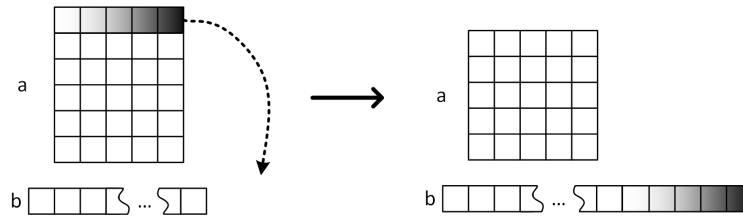


## Задача 7. Минимизация инверсий

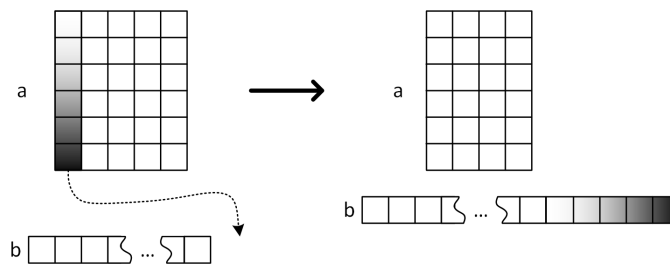
Ограничение по времени: 1 секунда  
Ограничение по памяти: 1024 мегабайта

Дана таблица  $a$ , состоящая из  $r$  строк и  $c$  столбцов, в которой записаны в произвольном порядке все различные числа от 1 до  $r \cdot c$ . Элементы этой таблицы переносятся в изначально пустой массив  $b$ . Пока таблица непустая, над ней выполняется одно из двух действий:

- Дописать в конец массива элементы первой строки таблицы в порядке от элемента в первом столбце до элемента в последнем и удалить первую строку из таблицы.



- Дописать в конец массива элементы первого столбца таблицы в порядке от элемента в первой строке до элемента в последней и удалить первый столбец из таблицы.



Порядок действий требуется выбирать таким, чтобы количество инверсий в полученном массиве после применения всех операций было минимальным.

Инверсией называется такая пара индексов элементов массива  $1 \leq i < j \leq r \cdot c$ , что  $b_i > b_j$ .

### Формат входных данных

Первая строка содержит два целых числа  $r$  и  $c$  ( $r \leq c$ ,  $1 \leq r \cdot c \leq 2\,000\,000$ ) — количество строк и столбцов в таблице соответственно.

В следующих  $r$  строках содержится описание таблицы  $a$ . В  $i$ -й из них содержится  $c$  целых чисел  $a_{i1}, \dots, a_{ic}$  ( $1 \leq a_{ij} \leq r \cdot c$ ) — элементы матрицы  $a$ .

Гарантируется, что все числа в таблице  $a$  различны.

### Формат выходных данных

Выведите одно число — минимально возможное количество инверсий в массиве  $b$  после применения всех операций.

## Система оценивания

Подзадача	Баллы	Ограничения			Необходимые подзадачи
		$r$	$c$	$r \cdot c$	
1	15	$r + c \leq 14$			У
2	18	–		$r \cdot c \leq 500$	У, 1
3	5	Все строки и столбцы отсортированы в возрастающем порядке и $r \cdot c \leq 250\,000$			–
4	7	$r = 1$	–	$r \cdot c \leq 250\,000$	–
5	6	$r \leq 2$			4
6	2	$r \leq 20$			У, 1, 4, 5
7	10	$r, c \leq 100$		–	У, 1
8	2	–		$r \cdot c \leq 10\,000$	У, 1, 2, 7
9	1	$r \leq 100$	$c \leq 1000$	–	У, 1, 2, 7
10	1		$c \leq 2500$		У, 1, 2, 7, 9
11	1		$c \leq 5000$		У, 1, 2, 7, 9, 10
12	1		$c \leq 7500$		У, 1, 2, 7, 9–11
13	1		$c \leq 10\,000$		У, 1, 2, 7–12
14	4		$c \leq 15\,000$		У, 1, 2, 7–13
15	2		$c \leq 20\,000$		У, 1, 2, 7–14
16	3		$r, c \leq 200$		У, 1, 7
17	3	$r, c \leq 400$		У, 1, 7, 16	
18	4	$r, c \leq 600$		У, 1, 2, 7, 16, 17	
19	1	$r, c \leq 800$		У, 1, 2, 7, 16–18	
20	1	$r, c \leq 1000$		У, 1, 2, 7, 9, 16–19	
21	1	$r, c \leq 1200$		У, 1, 2, 7, 9, 16–20	
22	1	$r, c \leq 1400$		У, 1, 2, 7, 9, 16–21	
23	1	–	$r \cdot c \leq 100\,000$		У, 1, 2, 7–9, 16
24	1		$r \cdot c \leq 250\,000$		У, 1–10, 16, 17, 23
25	4		$r \cdot c \leq 500\,000$		У, 1–11, 16–18, 23, 24
26	1		$r \cdot c \leq 750\,000$		У, 1–12, 16–19, 23–25
27	1		$r \cdot c \leq 1\,000\,000$		У, 1–13, 16–20, 23–26
28	1		$r \cdot c \leq 1\,500\,000$		У, 1–14, 16–21, 23–27
29	1		–		У, 1–28

## Примеры

стандартный ввод	стандартный вывод
2 3 3 4 1 5 6 2	6
2 3 2 3 4 1 6 5	2

## Замечание

В первом примере минимальное число инверсий достигается при двукратном удалении первой строки. В результате массив  $b$  будет равен  $[3, 4, 1, 5, 6, 2]$ . Такой массив содержит 6 инверсий.

Во втором примере для достижения минимального числа инверсий можно сначала удалить первый столбец, а потом два раза удалить первую строку. В результате массив  $b$  будет равен  $[2, 1, 3, 4, 6, 5]$ . Такой массив содержит 2 инверсии.

## Задача 8. Жизнь программистов

Ограничение по времени: 2 секунды  
Ограничение по памяти: 1024 мегабайта

Новый сериал про жизнь программистов содержит  $n$  серий, пронумерованных от 1 до  $n$ . Телекомпания Сириус ТВ планирует показывать серии по очереди от первой до последней в течение  $k$  дней, каждый день показывая блок из одной или нескольких подряд идущих серий. Каждая серия будет показана ровно один раз.

По результатам тестовых просмотров маркетологи компании составили рейтинг серий:  $i$ -й серии сопоставлено число  $a_i$  от 1 до  $n$ , самая интересная серия получила рейтинг 1, а самая скучная — рейтинг  $n$ . Рейтинги различных серий различны, поэтому числа  $[a_1, a_2, \dots, a_n]$  образуют перестановку.

Пусть принято решение о том, в какой день какие серии будут показаны. Для каждого дня определим рейтинг этого дня, равный рейтингу самой скучной серии этого дня. Иначе говоря, пусть в  $j$ -й день показываются серии с  $l_j$  по  $r_j$ , тогда рейтинг этого дня  $b_j$  равен максимальному значению среди  $[a_{l_j}, a_{l_j+1}, \dots, a_{r_j}]$ .

Чтобы показ сериала был удачным, необходимо вовлечь зрителей в просмотр. Среди всех возможных способов разбить серии на  $k$  блоков по дням необходимо выбрать тот, в котором рейтинг первого дня как можно лучше:  $b_1$  минимально. Среди этих способов в свою очередь необходимо минимизировать рейтинг второго дня  $b_2$ , при выбранных значениях  $b_1$  и  $b_2$  — минимизировать  $b_3$ , и так далее. Таким образом, необходимо разбить показ серий на  $k$  блоков таким образом, чтобы лексикографически минимизировать последовательность  $[b_1, b_2, \dots, b_k]$ .

Вам необходимо ответить на  $q$  запросов, каждый из которых задаётся двумя числами:  $k$  и  $i$ . В качестве ответа на запрос необходимо вывести значение  $b_i$  — рейтинг  $i$ -го дня для оптимального способа показать сериал за  $k$  дней.

### Формат входных данных

В первой строке входных данных содержится два целых числа  $n$  и  $q$  ( $1 \leq n, q \leq 300\,000$ ) — количество серий и количество запросов соответственно.

Во второй строке входных данных содержатся  $n$  целых чисел  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ) — рейтинги серий. Гарантируется, что массив  $a$  является перестановкой целых чисел от 1 до  $n$ .

Следующие  $q$  строк содержат по два целых числа  $k$  и  $i$  ( $1 \leq i \leq k \leq n$ ) — параметры очередного запроса.

### Формат выходных данных

В  $q$  строках выведите ответ на каждый запрос, в том порядке, в котором они даны во входных данных.

### Примеры

стандартный ввод	стандартный вывод
7 4	3
6 4 2 3 1 7 5	7
7 4	1
1 1	3
4 2	
5 3	
3 1	3
2 3 1	
2 2	



## Замечание

Рассмотрим первый тест:

- При  $k = 7$  существует единственный способ показа: каждый день показывать по одной серии. Рейтинги серий по дням получаются  $[6], [4], [2], [3], [1], [7], [5]$ , откуда  $b = [6, 4, 2, 3, 1, 7, 5]$ , поэтому ответ на запрос  $k = 7$  и  $i = 4$  равен  $b_4 = 3$ .
- При  $k = 1$  существует единственный способ показа: показать все серии в первый день. Рейтинги серий по дням:  $[6, 4, 2, 3, 1, 7, 5]$ , откуда  $b = [7]$ , поэтому ответ на запрос  $k = 1$  и  $i = 1$  равен  $b_1 = 7$ .
- При  $k = 4$  оптимально в первый день показать четыре серии, а затем три дня показывать по одной серии. Рейтинги серий по дням:  $[6, 4, 2, 3], [1], [7], [5]$ , откуда  $b = [6, 1, 7, 5]$ , поэтому ответ на запрос  $k = 4$  и  $i = 2$  равен  $b_2 = 1$ .
- При  $k = 5$  оптимально в первый и последний день показать по две серии, а в остальные дни по одной. Рейтинги серий по дням:  $[6, 4], [2], [3], [1], [7, 5]$ , откуда  $b = [6, 2, 3, 1, 7]$ , поэтому ответ на запрос  $k = 5$  и  $i = 3$  равен  $b_3 = 3$ .

## Система оценивания

Подзадача	Баллы	Доп. ограничения		Необх. подзадачи
		$n$	дополнительно	
1	5	$n \leq 20$	–	У
2	8	–	$k = 2$	–
3	8	–	$k = 3$	–
4	4	–	Перестановка имеет вид $1, n, 2, n - 1, \dots$	–
5	8	$n \leq 200$	–	У, 1
6	7	$n \leq 3000$	–	У, 1, 5
7	5	–	Количество различных значений $k$ во всех запросах не больше 10	У, 2, 3
8	5	–	$i \leq 3$	–
9	10	–	Количество значений $i$ , таких что $a_i < a_{i+1}$ , не больше 20	У, 1
10	8	–	Количество значений $i$ , таких что $a_i > a_{i+1}$ , не больше 20	У, 1
11	12	–	Перестановка была выбрана случайно	–
12	10	$n \leq 10^5$	–	У, 1, 5, 6
13	10	–	–	У, 1–12