

## Общая информация по задачам первого тура

Задача	Тип задачи	Ограничения
1. Видеонаблюдение	стандартная	1 с, 512 МБ
2. Тайное послание	с двойным запуском	1 с, 512 МБ
3. Рекорды и антирекорды	стандартная	3 с, 512 МБ
4. Ультра тех	стандартная	3 с, 512 МБ

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

Баллы за подзадачу, если в условии не указано иное, начисляются только если все тесты этой подзадачи пройдены. Решение запускается на тестах для определенной подзадачи, если все тесты всех необходимых подзадач пройдены.

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

## Задача 1. Видеонаблюдение

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

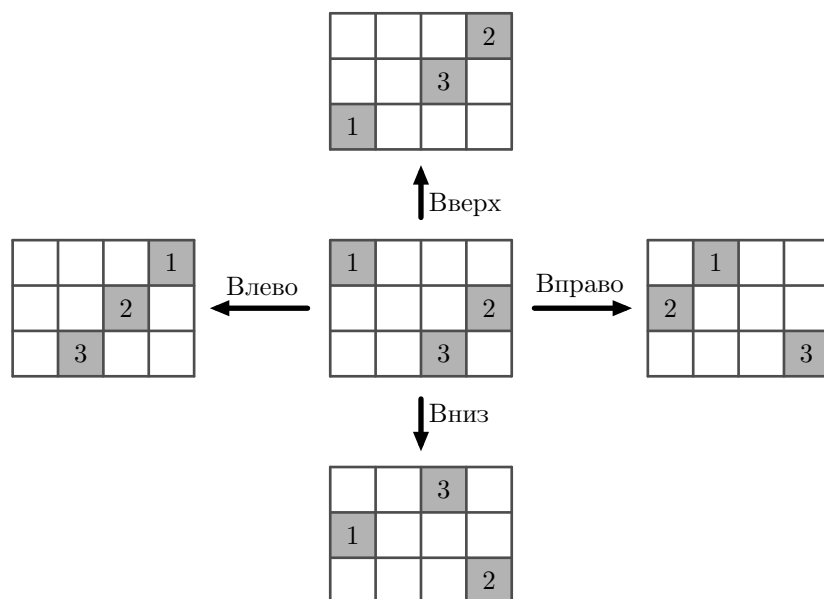
Безопасность в здании торгового центра обеспечивается с помощью системы видеонаблюдения. На компьютере у охранника открыта программа, которая выводит на экран видеопотоки с нескольких камер. Эта программа устроена следующим образом: на экране представлена прямоугольная сетка, состоящая из  $h$  строк и  $w$  столбцов. Каждая из ячеек может быть пуста, либо туда выводится изображение с одной из камер. Для управления расположением изображений в программе, сотрудник службы безопасности может использовать кнопки «влево», «вправо», «вверх» и «вниз».

Кнопка «влево» перемещает изображение из каждой ячейки в ячейку, находящуюся слева от нее. При этом, изображение из самой левой ячейки в каждом ряду перемещается в самую правую ячейку этого ряда.

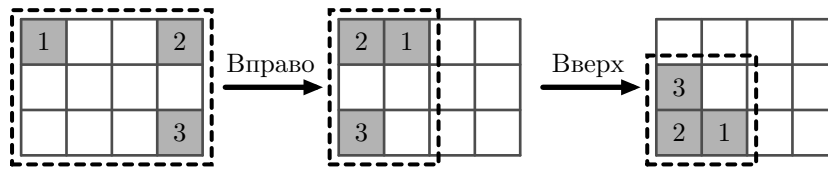
Аналогичным образом действуют кнопки «вправо», «вверх» и «вниз». Кнопка «вправо» перемещает изображение из каждой ячейки в ячейку, находящуюся справа от нее. Изображения из самой правой ячейки в каждом ряду перемещаются в самую левую ячейку этого ряда. Кнопка «вверх» перемещает изображение из каждой ячейки в ячейку, находящуюся над ней. Изображения из самого верхнего ряда перемещаются в ячейки самого нижнего ряда. Кнопка «вниз» перемещает изображение из каждой ячейки в ячейку, находящуюся под ней. Изображения из самого нижнего ряда перемещаются в ячейки самого верхнего ряда.

Строки в сетке пронумерованы сверху вниз, столбцы пронумерованы слева направо. Ячейка на пересечении строки номер  $r$  и столбца номер  $c$  обозначается как  $(r, c)$ .

Ниже изображена сетка с 3 строками, 4 столбцами и тремя ячейками, содержащими изображения, с координатами  $(1, 1)$ ,  $(2, 4)$  и  $(3, 3)$ . А также изображено, куда перейдут эти изображения при нажатии на каждую из четырёх кнопок.



Охраннику удобнее, чтобы ячейки на мониторе, которые содержат изображение с камер, были расположены как можно компактнее. Компактностью изображений назовём минимальную площадь подпрямоугольника сетки, который содержит все показываемые изображения. Заметим, что с помощью кнопок можно изменять компактность. Например, в левой части рисунка ниже показано расположение изображений, которое имеет компактность 12. Если один раз нажать на кнопку «вправо» и один раз нажать на кнопку «вверх», компактность изображений станет равна 4.



Вам дана сетка, которая содержит  $k$  ячеек с изображениями. Вычислите минимальную компактность, которую можно достичь, используя кнопки «влево», «вправо», «вверх» и «вниз», а также минимальное необходимое для этого количество нажатий на кнопки.

### Формат входных данных

В первой строке даны три целых числа  $h$ ,  $w$  и  $k$  — размеры сетки и количество ячеек с изображением ( $1 \leq h, w \leq 10^9$ ;  $1 \leq k \leq 100\,000$ ).

В каждой из следующих  $k$  строк даны по два целых числа  $r_i$  и  $c_i$  — координаты ячейки, содержащей изображение ( $1 \leq r_i \leq h$ ;  $1 \leq c_i \leq w$ ). Гарантируется, что все  $k$  ячеек различны.

### Формат выходных данных

Выведите два целых числа — минимальную компактность расположения изображений, которой можно достичь, используя кнопки, и минимально необходимое для достижения этой компактности количество нажатий на кнопки.

### Система оценки

Подзадача	Баллы	Дополнительные ограничения		Необх. подзадачи	Информация о проверке
		$n, m$	$k$		
1	5		$k = 1$		первая ошибка
2	10		$k = 2$		первая ошибка
3	29	$h = 1$			первая ошибка
4	11	$h, w \leq 50$		У	первая ошибка
5	15	$h, w \leq 1\,000$		У, 4	первая ошибка
6	6	$h, w \leq 200\,000$		У, 4–5	первая ошибка
7	24	Без дополнительных ограничений		У, 1–6	первая ошибка

### Примеры

стандартный ввод	стандартный вывод
1 10 3 1 5 1 7 1 2	6 0
3 4 3 1 1 3 4 1 4	4 2

## Задача 2. Тайное послание

Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Это задача с двойным запуском. На каждом тесте ваше решение будет запущено два раза.

На уроке информатики Алеся и Борис изучают криптографию. Ребята решили изобрести свой способ шифрования сообщений.

Алеся выбирает  $k$  различных целых чисел от 1 до  $n$  и обозначает получившееся множество как  $T$ . Алеся хочет передать Борису в качестве сообщения множество  $T$  в зашифрованном виде. Для этого по множеству  $T$  Алеся построит и передаст Борису другое множество  $R$ , также состоящее из целых чисел от 1 до  $n$ .

Ребята не хотят, чтобы после шифрования размер сообщения изменялся, поэтому  $R$  также должно содержать ровно  $k$  чисел. А ещё они считают, что если  $T$  и  $R$  будут содержать хотя бы один общий элемент, то их шифрование будет недостаточно надёжным. Поэтому не должно существовать числа, которое входит и в  $T$ , и в  $R$ , то есть множества  $T$  и  $R$  не должны пересекаться. Гарантируется, что  $k \leq n/2$ , поэтому по множеству  $T$  всегда возможно построить хотя бы одно множество  $R$ .

Когда Борис получит зашифрованное сообщение  $R$ , он должен будет его расшифровать и получить исходное сообщение  $T$ .

Помогите Алесе и Борису придумать и реализовать алгоритмы шифрования и дешифрования. При первом запуске ваша программа будет выступать в роли Алеси, а при втором запуске — в роли Бориса.

### Формат входных данных

В первой строке входных данных дано одно число  $a$ , равное 1 или 2 — номер запуска вашей программы.

Во второй строке дано одно число  $m$  — количество сообщений ( $1 \leq m \leq 300\,000$ ), которое ваша программа должна зашифровать (в первом запуске) или расшифровать (во втором запуске).

Следующие  $2m$  строк содержат описания  $m$  сообщений, по две строки на сообщение.

В первой строке сообщения записаны два целых числа  $n_i$  и  $k_i$  ( $2 \leq n_i \leq 10^9$ ,  $1 \leq k_i \leq 300\,000$ ,  $k_i \leq \frac{n_i}{2}$ ). Во второй строке сообщения записаны  $k_i$  различных целых чисел от 1 до  $n_i$  в возрастающем порядке.

Гарантируется, что сумма всех значений  $k_i$  в одном тесте не превосходит 300 000.

Если  $a = 1$ , то данные числа являются исходным сообщением. Если  $a = 2$ , то данные числа являются результатом запуска вашей программы для шифрования какого-либо сообщения при первом запуске вашей программы.

### Формат выходных данных

Программа должна вывести  $m$  строк,  $i$ -я строка должна содержать  $k_i$  различных целых чисел от 1 до  $n_i$  в возрастающем порядке.

При первом запуске для каждого исходного сообщения  $T_i$  программа должна вывести множество  $R_i$ , которое не должно пересекаться с  $T_i$ .

При втором запуске программа для каждого зашифрованного сообщения  $R_i$  должна восстановить исходное сообщение  $T_i$ .

### Пример

Обратите внимание, что в примере приведены конкретные варианты вывода в первом запуске и ввода во втором запуске. Если ваша программа выведет другое множество  $R$ , при втором запуске ввод также будет другой.

Также при втором запуске зашифрованные сообщения передаются программе участника не обязательно в том порядке, в котором они следовали при первом запуске.

Первый запуск.

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

Второй запуск.

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

## Система оценки

Чтобы указать дополнительные ограничения на входные данные, обозначим последовательность чисел, которые задают множество  $T_i$ , как  $t_1, t_2, \dots, t_{k_i}$ . Они расположены в порядке возрастания.

Обозначим сумму  $n_i$  в одном тесте как  $N$ .

Обозначим сумму  $k_i$  в одном тесте как  $K$ .

Подз.	Баллы	Дополнительные ограничения			Необх. подзадачи	Информация о проверке
		$n_i, N$	$k_i, K$	$t_j$		
1	9	$N \leq 5\,000$	$k_i = 1$			первая ошибка
2	11	$N \leq 5\,000$	$k_i = 2$			первая ошибка
3	9	$N \leq 300\,000$	$k_i = \frac{n_i}{2}$			первая ошибка
4	7	$n_i \leq 7$ $N \leq 5\,000$			У	первая ошибка
5	9			$t_{j+1} - t_j \geq 2$ $t_{k_i} \leq n_i - 1$		первая ошибка
6	9			$t_{k_i} - t_1 \leq \frac{n_i}{2} - 1$		первая ошибка
7	10	$N \leq 5\,000$		$t_{k_i} \leq n_i - k_i$		первая ошибка
8	12	$N \leq 100$			У	первая ошибка
9	3	$N \leq 5\,000$			У, 1-2, 4, 7-8	первая ошибка
10	7	$N \leq 300\,000$			У, 1-4, 7-9	первая ошибка
11	7		$K \leq 5\,000$		У, 1-2, 4, 7-9	первая ошибка
12	7	Без дополнительных ограничений			У, 1-11	первая ошибка

## Задача 3. Рекорды и антирекорды

Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

Перестановкой размера  $n$  называется последовательность  $n$  целых чисел  $a_1, a_2, \dots, a_n$ , в которой все значения от 1 до  $n$  встречаются ровно по одному разу.

Последовательность  $b_1, b_2, \dots, b_l$  является подпоследовательностью последовательности  $a_1, a_2, \dots, a_n$ , если  $b$  можно получить из  $a$  удалением некоторых элементов (то есть  $l \leq n$  и существуют  $i_1 < i_2 < \dots < i_l$  такие что  $a_{i_t} = b_t$ ).

- Элемент последовательности  $a_i$  называется *рекордом*, если он строго больше всех предыдущих элементов (то есть  $a_j < a_i$  для всех  $1 \leq j < i$ ).
- Элемент последовательности  $a_i$  называется *антирекордом*, если он строго меньше всех предыдущих элементов (то есть  $a_j > a_i$  для всех  $1 \leq j < i$ ).

Дана перестановка  $p_1, p_2, \dots, p_n$  размера  $n$ . Требуется разделить её на две непустые подпоследовательности  $q$  и  $r$ . Иными словами, каждый элемент  $p$  должен попасть ровно в одну из подпоследовательностей. При этом требуется максимизировать сумму количества рекордов в  $q$  и количества антирекордов в  $r$ .

### Формат входных данных

Каждый тест состоит из нескольких наборов входных данных. В первой строке содержится единственное целое число  $t$  ( $1 \leq t \leq 100\,000$ ) — количество наборов входных данных. В следующих  $2t$  строках следуют описания наборов входных данных.

В первой строке описания каждого набора входных данных содержится одно целое число  $n$  — размер перестановки ( $2 \leq n \leq 200\,000$ ).

Во второй строке описания каждого набора входных данных содержатся  $n$  целых чисел  $p_1, p_2, \dots, p_n$  — исходная перестановка. Гарантируется, что среди элементов  $p$  каждое число от 1 до  $n$  встречается ровно по одному разу.

Сумма  $n$  по всем наборам входных данных не превосходит 200 000.

### Формат выходных данных

Для каждого набора входных данных выведите одно целое число — максимальную возможную сумму количества рекордов в  $q$  и антирекордов в  $r$  в оптимальном разбиении.

### Система оценки

Обозначим как  $N$  сумму значений  $n$  во всех наборах входных данных одного теста.

Подз.	Баллы	Ограничения		Необх. подзадачи	Информация о проверке
		$n, N$	Доп. ограничения		
1	21	$n \leq 16$	$t \leq 120$	У	первая ошибка
2	22	$n \leq 200,$ $N \leq 2000$		У, 1	первая ошибка
3	14	$N \leq 2\,000$		У, 1–2	первая ошибка
4	10	$N \leq 10\,000$		У, 1–3	первая ошибка
5	13	$N \leq 200\,000$	Длина наибольшей убывающей подпоследовательности в $p$ не превышает 2		первая ошибка
6	20	$N \leq 200\,000$		У, 1–5	первая ошибка

## Пример

стандартный ввод	стандартный вывод
4	5
5	6
4 1 2 3 5	3
10	5
3 8 10 4 1 2 7 9 5 6	
3	
1 2 3	
6	
4 2 5 1 6 3	

## Замечание

Один из способов оптимальным образом разбить  $p$  на  $q$  и  $r$  в первом наборе входных данных (рекорды в  $q$  и антирекорды в  $r$  обведены):

- $q = \boxed{1} \boxed{2} \boxed{3} \boxed{5}$
- $r = \boxed{4}$

Один из способов оптимальным образом разбить  $p$  на  $q$  и  $r$  во втором наборе входных данных:

- $q = \boxed{3} \boxed{8} 4 1 2 \boxed{9}$
- $r = \boxed{10} \boxed{7} \boxed{5} 6$

## Задача 4. Ультра мех

Ограничение по времени: 3 секунды  
Ограничение по памяти: 512 мегабайт

Рассмотрим  $A$  — множество неотрицательных целых чисел. Минимальное неотрицательное целое число, которое не встречается в  $A$ , обозначим как  $\text{mex}(A)$ . Например,  $\text{mex}(\{0, 1, 2, 4, 5, 9\}) = 3$ . Эта функция часто используется, например, в теории игр.

Операция «побитовое исключающее или» (обозначается «xor» в Паскале, «^» в C++, Python и Java) для двух целых чисел определена следующим образом:  $i$ -й бит результата равен 1 тогда и только тогда, когда в одном из чисел этот бит 1, а в другом 0. Будем обозначать эту операцию символом  $\oplus$ . Например,  $6 \oplus 10 = 110_2 \oplus 1010_2 = 1100_2 = 12$ .

Определим ещё одну операцию над множеством  $A$ , содержащим число 0. Операция будет называться «ультра». Пусть  $m = \text{mex}(A)$ . Заметим, что  $m > 0$ . Построим новое множество  $\text{ultra}(A)$  следующим образом: применим «побитовое исключающее или» с числом  $(m-1)$  ко всем элементам  $A$ . Например,  $\text{ultra}(\{0, 1, 2, 4, 5, 9\}) = \{0 \oplus 2, 1 \oplus 2, 2 \oplus 2, 4 \oplus 2, 5 \oplus 2, 9 \oplus 2\} = \{2, 3, 0, 6, 7, 11\} = \{0, 2, 3, 6, 7, 11\}$ . Можно показать, что если множество  $A$  содержит 0, то множество  $\text{ultra}(A)$  также содержит 0.

Выберем множество  $A_0$ , состоящее из целых чисел от 0 до  $2^k - 1$  и содержащее 0. Рассмотрим следующую последовательность:

- $m_0 = \text{mex}(A_0)$ ,  $A_1 = \text{ultra}(A_0)$
- $m_1 = \text{mex}(A_1)$ ,  $A_2 = \text{ultra}(A_1)$
- ...
- $m_i = \text{mex}(A_i)$ ,  $A_{i+1} = \text{ultra}(A_i)$
- ...

Будем называть множество  $A_0$  *мех-стабильным*, если начиная с некоторого индекса  $l$  числа  $m_i$  перестают меняться. То есть, для всех  $i \geq l$  выполнено  $m_i = m_l$ . Число  $m_l$  будем называть *мех-пределом* множества  $A_0$ .

Вам даны числа  $k$ ,  $n$  и  $p$ . Вычислите количество множеств  $A_0$ , которые:

- Состоят из  $n$  различных чисел от 0 до  $2^k - 1$  (0 обязательно должен входить в  $A_0$ );
- Являются мех-стабильными;
- мех-предел  $A_0$  равен  $p$ .

Так как ответ может быть большим, выведите его по простому модулю  $M$ . Гарантируется, что  $(M-1)$  делится на  $2^{18}$ .

### Формат входных данных

В первой строке дано одно целое число  $M$  — модуль, по которому нужно посчитать ответ ( $3 \leq M \leq 10^9$ ;  $(M-1)$  делится на  $2^{18}$ ). Гарантируется, что  $M$  простое число.

Во второй строке дано одно целое число  $t$  — количество наборов входных данных ( $1 \leq t \leq 100\,000$ ).

Для каждого набора входных данных в единственной строке даны три целых числа  $k$ ,  $n$  и  $p$  ( $1 \leq k \leq 17$ ;  $1 \leq n, p \leq 2^k$ ).

### Формат выходных данных

Для каждого набора входных данных на новой строке выведите одно целое число — количество искоемых множеств  $A$ , взятое по модулю  $M$ .



## Система оценки

В этой задаче 30 подзадач. В каждой подзадаче показывается первая ошибка.

В таблице ниже приведены ограничения на  $k$  и  $t$  в каждой подзадаче. Для каждой подзадачи необходимыми являются все другие подзадачи с не большими ограничениями на  $k$  и  $t$ .

$k$	$t \leq 10$		$t \leq 10^5$	
	Номер	Баллы	Номер	Баллы
$k \leq 1$	1	3	–	–
$k \leq 2$	2	5	–	–
$k \leq 3$	3	7	–	–
$k \leq 4$	4	8	–	–
$k \leq 5$	5	3	6	3
$k \leq 6$	7	3	8	3
$k \leq 7$	9	3	10	3
$k \leq 8$	11	2	12	2
$k \leq 9$	13	2	14	2
$k \leq 10$	15	3	16	3
$k \leq 11$	17	3	18	3
$k \leq 12$	19	3	20	3
$k \leq 13$	21	3	22	3
$k \leq 14$	23	3	24	3
$k \leq 15$	25	4	26	3
$k \leq 16$	27	4	28	3
$k \leq 17$	29	4	30	3

## Пример

стандартный ввод	стандартный вывод
998244353	6
6	1
3 2 1	0
3 2 2	0
3 2 3	29
3 2 4	2461
3 5 1	
4 6 1	

## Замечание

Всего существует 7 мех-стабильных множеств размера 2 из чисел от 0 до 7:  $\{0, 1\}$ ,  $\{0, 2\}$ ,  $\{0, 3\}$ ,  $\{0, 4\}$ ,  $\{0, 5\}$ ,  $\{0, 6\}$ ,  $\{0, 7\}$ .

Для  $\{0, 1\}$ :  $\text{mex}(\{0, 1\}) = 2$ ,  $\text{ultra}(\{0, 1\}) = \{0 \oplus 1, 1 \oplus 1\} = \{1, 0\} = \{0, 1\}$ , получается, что  $A_1 = A_0$ . Значит мех-предел будет равен 2.

Для всех остальных множеств  $m_0 = \text{mex}(A_0) = 1$ , для них при вычислении  $\text{ultra}$  происходит  $\oplus$  с числом 0, поэтому  $\text{ultra}(A_0) = A_0$ . Получается, для них мех-предел равен  $\text{mex}(A_0) = 1$ .