

## Задача А. Re: Fwd: Про чайник

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

После окончания карантина Саша планирует встретиться с друзьями и выбраться на пикник. Саша очень любит чай, а для этого на пикник придется взять чайник и воду. Саша пригласила на пикник  $m$  друзей и взяла с собой  $n$  бутылок воды, каждая из которых содержит ровно  $a$  литров воды.

Перед использованием чайник необходимо помыть, для этого требуется  $x$  литров воды. После этого Саша хочет налить каждому из приглашенных  $m$  друзей чай: Саша использует для этого оставшуюся воду, при этом, она хочет, чтобы каждому из друзей досталось одинаковое целое число литров воды. Помогите Саше выяснить, какое максимальное количество литров воды может достаться каждому из друзей.

### Формат входных данных

В первой строке вводится число  $n$  ( $0 \leq n \leq 10^9$ ) — количество бутылок, которые друзья взяли с собой.

Во второй строке вводится число  $a$  ( $0 \leq a \leq 10^9$ ) — количество литров воды, которые вмещает одна бутылка.

В третьей строке вводится число  $x$  ( $0 \leq x \leq 10^{18}$ ) — количество литров воды, которое необходимо, чтобы промыть чайник.

В четвёртой строке вводится число  $m$  ( $1 \leq m \leq 10^{18}$ ) — количество людей, которые пойдут на пикник.

### Формат выходных данных

Выведите одно целое число — количество литров воды, которое достанется каждому из друзей.

### Примеры

стандартный ввод	стандартный вывод
3 6 5 2	6
2 6 10 5	0

### Замечание

В первом примере для чистки чайника будет вылито 5 литров воды из первой бутылки. После этого всю воду из второй бутылки можно вскипятить и дать первому человеку, а всю воду из третьей бутылки — второму. После этого останется ещё один литр воды в первой бутылке, но так как каждому человеку надо дать целое число литров воды, то этот литр нельзя будет разделить на двоих.

### 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.

Длинный тур отборочного этапа Открытой олимпиады школьников 2020–2021 учебного года  
Россия, 16 ноября 2020 года – 20 января 2021 года

Группа	Баллы	Дополнительные ограничения				Необх. группы	Комментарий
		$n$	$a$	$x$	$m$		
0	0	–	–	–	–	–	Тесты из условия.
1	9	$n \leq 10$	$a \leq 10$	$x \leq 10$	$m \leq 10$	0	
2	15	$n \leq 500$	$a \leq 500$	$x \leq 500$	$m \leq 500$	0, 1	
3	12	$n \leq 10^4$	$a \leq 10^4$	$x = 0$	$m \leq 10^9$	–	
4	11	$n \leq 10^4$	$a \leq 10^4$	$x \leq 10^9$	$m = 1$	–	
5	18	$n \leq 10^4$	$a \leq 10^4$	$x \leq 10^9$	$m \leq 10^9$	0–4	
6	35	–	–	–	–	0–5	

## Задача В. Проверка олимпиады

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

На национальную олимпиаду по математике приехало  $n$  участников со всей страны, каждый из которых сдал свою работу. Теперь перед жюри стоит непростая задача: проверить все  $n$  работ как можно быстрее.

Процесс проверки происходит следующим образом. Каждый из  $m$  проверяющих может выбрать одну из работ участников, которую он будет проверять в течение одного часа. Два проверяющих не могут одновременно проверять одну и ту же работу, поэтому различные проверяющие должны выбрать различные работы. Некоторые проверяющие могут ничего не выбрать. Через час проверка оканчивается, и работы заново распределяются между проверяющими.

Чтобы сделать проверку более объективной, жюри придумало следующее правило: каждую работу должны проверить как минимум  $k$  разных людей.

Помогите жюри составить подходящее расписание проверки, занимающее минимальное время.

### Формат входных данных

На единственной строке даны три целых числа  $n$ ,  $m$ ,  $k$  ( $1 \leq k \leq m \leq n \leq 500$ ) – количество работ, общее число проверяющих и минимальное число людей, которые должны проверить каждую работу.

### Формат выходных данных

В первой строке выведите единственное целое число  $t$  – минимальное число часов, которое потребуется, чтобы каждую работу проверило хотя бы  $k$  проверяющих.

Далее выведите  $t$  строк по  $n$  чисел  $C_{ij}$  ( $0 \leq C_{ij} \leq m$ ), обозначающих номер проверяющего работы номер  $j$  во время часа  $i$ .  $C_{ij} = 0$  означает, что работа  $j$  в час  $i$  не проверяется.

Если возможно несколько вариантов ответа, выведите любой из них.

### Примеры

стандартный ввод	стандартный вывод
3 1 1	3 1 0 0 0 1 0 0 0 1
4 2 2	4 1 2 0 0 0 0 1 2 2 1 0 0 0 0 2 1
5 3 2	4 1 2 3 0 0 3 0 0 1 2 0 1 2 3 0 0 0 0 0 1

### Система оценки

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

Группа	Баллы	Доп. ограничения			Необх. группы	Комментарий
		$n$	$m$	$k$		
0	0	–	–	–	–	Тесты из условия.
1	7	$n \leq 4$	$m \leq 4$	$k \leq 4$	–	
2	9	$n \leq 9$	$m \leq 9$	$k \leq 9$	0, 1	
3	8	–	$m = 1$	$k = 1$	–	
4	13	–	–	–	–	$n = m$
5	17	–	–	–	3, 4	$n$ делится на $m$ .
6	11	–	–	$k = 1$	3	
7	35	–	–	–	0–6	

## Задача С. Дед и мопед

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 512 мегабайт

Дед Максим собирается в путешествие по Флатландии. К сожалению, из доступных средств передвижения у него есть только мопед, запас хода которого ограничен. Более точно, если бак мопеда полностью заполнен, то мопед может проехать не более  $s$  километров без дополнительной дозаправки.

Всего во Флатландии есть  $n$  городов, пронумерованных от 1 до  $n$ . В некоторых городах находятся заправки, и если в городе есть заправка, то в этом городе дед Максим может полностью наполнить бак. К сожалению, заправки присутствуют лишь в  $k$  городах. Также во Флатландии есть  $m$  дорог,  $i$ -я из которых соединяет города  $u_i$  и  $v_i$  и имеет длину  $c_i$  километров. По каждой дороге можно перемещаться в обоих направлениях.

Дед Максим начинает свое путешествие в городе с номером 1 с полным баком (в городе 1 есть заправка). Помогите ему определить, до каких городов он сможет добраться на мопеде.

### Формат входных данных

В первой строке записаны четыре целых числа  $n, m, k, s$  ( $1 \leq n \leq 150\,000$ ,  $0 \leq m \leq 150\,000$ ,  $1 \leq k \leq n$ ,  $1 \leq s \leq 10^9$ ) — количество городов, количество дорог, количество заправок и объем бака.

В следующих  $m$  строках записаны по три целых числа  $u_i, v_i, c_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ,  $1 \leq c_i \leq s$ ) — начало, конец и длина  $i$ -й дороги. Гарантируется, что не существует двух дорог, соединяющих одинаковую пару городов.

В следующей строке записаны  $k$  целых чисел  $p_i$  ( $1 \leq p_i \leq n$ ) — номера городов с заправками. Гарантируется, что заправка присутствует в городе 1.

### Формат выходных данных

В первой строке выведите единственное число  $x$ : количество городов, до которых дед Максим может добраться.

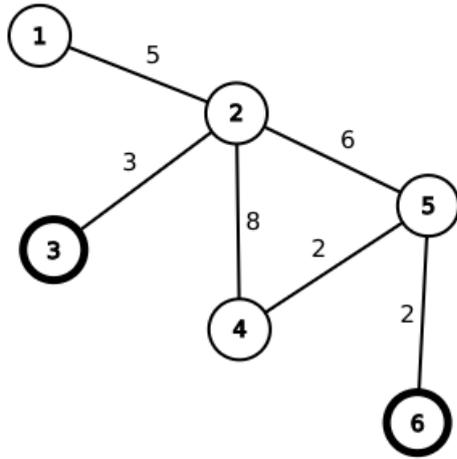
Во второй строке выведите  $x$  целых чисел — номера подходящих городов **в порядке возрастания**.

### Пример

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

### Замечание

Рисунок ниже иллюстрирует первый пример из условия:



Из начального города с номером 1 можно доехать до городов с номерами 2 и 3 без дополнительных дозаправок. Также можно доехать до города с номером 3, пополнить там бак и доехать до города с номером 5. До городов 4 и 6 добраться невозможно, так как минимальное расстояние от достижимой заправки до них равно 11.

### Система оценки

Тесты к этой задаче состоят из семи 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
		$k$	$n$	$c_i$		
0	0	–	–	–	–	Тесты из условия.
1	15	–	$n \leq 100$	–	0	
2	11	$k = 1$	$n \leq 5000$	–	–	
3	8	$k = 1$	–	–	2	
4	12	–	$n \leq 5000$	$c_i = 1$	–	
5	9	–	$n \leq 5000$	$c_i \leq 10$	0, 4	
6	17	–	$n \leq 5000$	–	0, 1, 2, 4, 5	
7	28	–	–	–	0–6	<b>Offline-проверка.</b>

## Задача D. Задача для разминки рук

Имя входного файла: стандартный ввод или `input.txt`  
Имя выходного файла: стандартный вывод или `output.txt`  
Ограничение по времени: 6.5 секунд  
Ограничение по памяти: 768 мегабайт

Дано дерево на  $n$  вершинах с корнем в вершине 1. В  $i$ -й вершине записан символ  $t_i$  — одна из трёх латинских букв `a`, `b`, `c`. Также дана строка  $s$  длины  $m$ , состоящая из строчных латинских букв `a`, `b`, `c`.

Каждой паре вершин  $(u, v)$  естественным образом можно сопоставить строку, которая получается последовательным выписыванием символов в вершинах на единственном простом пути от  $u$  до  $v$ , начиная с символа в вершине  $u$ .

Вам нужно посчитать количество пар целых чисел  $(u, v)$  таких, что  $1 \leq u, v \leq n$ , и строка, соответствующая пути от  $u$  до  $v$ , лексикографически меньше или равна  $s$ .

Деревом называется связный граф, где у каждой вершины кроме первой (первая вершина называется корнем) задан единственный предок, с которым вершина соединена ребром. Путём в дереве между вершинами  $u$  и  $v$  называется такая последовательность вершин, в которой  $u$  — первая вершина,  $v$  — последняя, и любые 2 подряд идущие вершины соединены ребром. Путь называется простым, если никакая из вершин не встречается в нем дважды.

Строка  $a$  считается лексикографически меньше строки  $b$ , если существует такое число  $k$ , что на всех позициях меньших  $k$  строки  $a$  и  $b$  совпадают, а на  $k$ -й позиции символ строки  $a$  лексикографически меньше соответствующего символа строки  $b$ , или же в том случае, если длина строки  $a$  меньше длины строки  $b$ , а все символы на одинаковых позициях у строк совпадают.

### Формат входных данных

В первой строке вводятся два целых числа  $n$  и  $m$  ( $2 \leq n \leq 10^6$ ,  $1 \leq m \leq 10^6$ ) — число вершин в дереве и длина строки для сравнения.

Во второй строке без пробелов вводятся  $m$  символов  $s_1s_2 \dots s_m$  — строка для сравнения.

В третьей строке без пробелов вводятся  $n$  символов  $t_1t_2 \dots t_n$ ,  $i$ -й из которых обозначает букву, записанную в вершине  $i$ .

В следующей строке вводится  $(n - 1)$  число  $p_2, p_3, p_4 \dots, p_n$  ( $1 \leq p_i \leq n$ ), где  $p_i$  обозначает номер предка вершины  $i$  в дереве.

Гарантируется, что каждый символ в данных строках  $s$  и  $t$  равен одной из строчных латинских букв «a», «b», «c», и что заданный граф действительно образует дерево.

### Формат выходных данных

Выведите единственное число — искомое количество пар.

### Примеры

стандартный ввод	стандартный вывод
3 2 ba acb 3 1	5
5 3 bac abcab 1 1 3 1	18
2 3 acc bb 1	0

## Замечание

В первом примере всего есть 9 пар чисел от 1 до 3. Парам (1, 1), (1, 2), (1, 3), (3, 3), (3, 1) соответствуют строки **a**, **abc**, **ab**, **b**, **ba**. Парам, где первое число 2, соответствуют строки, начинающиеся на **c** — эти строки в любом случае лексикографически больше строки **ba**. Паре (3, 2) соответствует строка **bc**, которая лексикографически больше **ba**.

Во втором примере всего есть 25 пар. Первым числом в паре не может быть 3, так как в вершине 3 записана буква **c**, которая меньше первой буквы **s**. Также не подходят пары (2, 4) и (5, 4).

В третьем примере все символы в вершинах больше первой буквы **s**, поэтому ни одна пара вершин не подходит.

## 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.

Группа	Баллы	Доп. ограничения		Необх. группы	Комментарий
		$n$	$m$		
0	0	–	–	–	Тесты из условия.
1	11	$n \leq 500$	$m \leq 500$	0	
2	8	$n \leq 5000$	$m \leq 5000$	0, 1	
3	13	$n \leq 30\,000$	$m \leq 500$	0, 1	Нет путей длины больше 500.
4	10	$n \leq 100\,000$	$m \leq 100\,000$	–	$p_i = i - 1$
5	14	$n \leq 100\,000$	$m \leq 100\,000$	–	$s$ и $t$ состоят из символов « <b>a</b> ».
6	8	$n \leq 100\,000$	$m \leq 100\,000$	0–5	
7	5	$n \leq 200\,000$	$m \leq 200\,000$	0–6	
8	5	$n \leq 350\,000$	$m \leq 350\,000$	0–7	
9	6	$n \leq 500\,000$	$m \leq 500\,000$	0–8	
10	6	$n \leq 650\,000$	$m \leq 650\,000$	0–9	<b>Offline-проверка.</b>
11	7	$n \leq 800\,000$	$m \leq 800\,000$	0–10	<b>Offline-проверка.</b>
12	7	–	–	0–11	<b>Offline-проверка.</b>

## Задача Е. Проблема останова

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:	512 мегабайт

Проблема останова — это одна из ключевых задач в теории вычислимости. Её формулировка такова: по данному тексту программы и входным данным необходимо определить, остановится ли программа на этих входных данных. В этой задаче вам предлагается решить её расширенную версию: по данному тексту программы определить, останавливается ли она на всех возможных входных данных.

Программа написана на простом языке, который может оперировать  $k$  регистрами, каждый из которых содержит 0 или 1. Регистры нумеруются с 0. Язык поддерживает следующие команды:

- `begin` — начало программы. Программа всегда начинается с этой команды.
- `and to from` — применить логическое «И» к значениям в регистрах с номерами `to` и `from` и сохранить полученное значение в регистр с номером `to`.
- `or to from` — применить логическое «ИЛИ» к значениям в регистрах с номерами `to` и `from` и сохранить полученное значение в регистр с номером `to`.
- `xor to from` — применить логическое «исключающее ИЛИ» к значениям в регистрах с номерами `to` и `from` и сохранить полученное значение в регистр с номером `to`.
- `move to from` — скопировать значение из регистра с номером `from` в регистр номер `to`. Значение регистра с номером `from` остаётся неизменным.
- `set to x` — присвоить регистру с номером `to` значение  $x$  ( $x \in \{0, 1\}$ ).
- `not to` — применить логическое «НЕ» к значению в регистре с номером `to`.
- `jump reg line` — если в регистре с номером `reg` содержится 1, то перейти на строку программы с номером `line`, иначе перейти на следующую строку. Строки программы нумеруются с 0.
- `end` — конец программы. Программа всегда заканчивается этой командой.

Входные данные состоят из  $k$  битов и в начале исполнения лежат в регистрах. По данному тексту программы определите, останавливается ли она на всех входных данных, а если существуют такие входные данные, на которых программа будет работать вечно, выведите их.

### Формат входных данных

В первой строке входных данных содержатся два числа  $n$  и  $k$  ( $2 \leq n \leq 1000$ ,  $1 \leq k \leq 12$ ) — количество строк в программе и количество регистров, соответственно.

В следующих  $n$  строках содержится текст программы в формате, указанном в условии. Гарантируется, что он корректен, то есть `begin` содержится только в первой строке, `end` только в последней, все номера регистров находятся в пределах от 0 до  $k - 1$  включительно, второй аргумент команды `set` может быть равен только 0 или 1, а аргумент `line` команды `jump` находится в пределах от 0 до  $n - 1$  включительно.

### Формат выходных данных

В случае, если программа всегда останавливается, выведите `Yes`.

В противном случае, в первой строке выведите `No`. Во второй строке выведите последовательность из  $k$  чисел  $x_i$  ( $x_i \in \{0, 1\}$ ) без пробелов, где  $x_i$  равняется значению  $i$ -го регистра во входных данных, на которых программа не прекратит работу.

Если входных данных, на которых программа будет работать вечно, несколько, выведите любой пример.

## Примеры

стандартный ввод	стандартный вывод
3 6 begin not 0 end	Yes
8 3 begin set 1 1 xor 0 1 xor 0 1 or 0 2 and 0 1 jump 0 6 end	No 100
5 6 begin move 2 1 move 3 2 jump 3 1 end	No 010000
7 6 begin set 0 1 or 1 0 jump 1 6 set 0 1 jump 0 4 end	Yes

## Замечание

В первом тестовом примере в программе нет ни одной команды `jump`, поэтому она завершится на любых входных данных.

Разберём работу программы из второго тестового примера на двух входных данных: 000 и 100.  
Для 000:

1. Присваиваем регистру номер 1 значение 1. Значения регистров: 010
2. Делаем логическое «исключающее ИЛИ» с регистрами номер 0 и 1, результат записываем в регистр 0. Значения регистров: 110
3. Делаем еще раз то же самое, значения регистров: 010
4. Делаем логическое «ИЛИ» с регистрами номер 0 и 2, результат записываем в регистр 0. Значения регистров: 010
5. Делаем логическое «И» с регистрами номер 0 и 1, результат записываем в регистр 0. Значения регистров: 010
6. Так как в регистре номер 0 хранится 0, не переходим на строку номер 6 (то есть на текущую), переходим на следующую. Значения регистров: 010
7. Дошли до последней строки с командой `end`, программа завершилась.

Для 100:

1. Присваиваем регистру номер 1 значение 1. Значения регистров: 110
2. Делаем логическое «исключающее ИЛИ» с регистрами номер 0 и 1, результат записываем в регистр 0. Значения регистров: 010
3. Делаем еще раз то же самое, значения регистров: 110
4. Делаем логическое «ИЛИ» с регистрами номер 0 и 2, результат записываем в регистр 0. Значения регистров: 110
5. Делаем логическое «И» с регистрами номер 0 и 1, результат записываем в регистр 0. Значения регистров: 110
6. Так как в регистре номер 0 хранится 1, переходим на строку номер 6 (то есть на текущую). После этого нам нужно будет снова выполнить эту строку, и таким образом мы вошли в бесконечный цикл.

В третьем тестовом примере проиллюстрирована работа `move`.

В четвёртом тестовом примере в тексте программы есть бесконечный цикл, но он недостижим, а программа останавливается на всех входных данных.

## Система оценки

Тесты к этой задаче состоят из четырёх 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.

Подзадача	Баллы	Доп. ограничения	Необходимые подзадачи	Комментарий
		$k$		
0	0	–	–	Тесты из условия.
1	18	–	–	Помимо <code>begin</code> и <code>end</code> в программе содержатся только команды <code>jump</code> .
2	17	$k = 1$	–	
3	28	$k \leq 6$	0, 2	
4	37	–	0–3	

## Задача F. Футболки на олимпиаду

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:	512 мегабайт

Наконец-то пришло время для финала Закрытой олимпиады школьников по программированию! Все отборы пройдены, сувенирные футболки лежат в пакетах. Вот только распределение участников по площадкам еще не спланировано.

У жюри олимпиады есть  $m$  площадок проведения. На  $i$ -й площадке лежит пакет, в котором находится  $a_i$  футболок. Это значит, что на эту площадку можно зарегистрировать не более  $a_i$  участников. Всего же среди финалистов будет  $n$  человек.

Финал Закрытой олимпиады школьников по программированию проходит в городе-герое Кленинграде. Как любой другой город Флатландии, он расположен в декартовой системе координат. Жюри знает адрес каждого финалиста — координаты  $X_i, Y_i$ , а также адреса площадок — координаты  $x_i, y_i$ .

Жюри хочет так распределить участников по точкам проведения, чтобы всем участникам хватило футболок, при этом максимальное из расстояний, которое придется пройти участнику до точки проведения, было как можно меньше. Поскольку жюри олимпиады сейчас везет на точки проведения другую сувенирку, Вас попросили провести распределение участников по площадкам.

### Формат входных данных

В первой строке заданы два целых числа  $n, m$  ( $1 \leq n, m \leq 500$ ) — количество участников олимпиады и точек проведения.

В следующих  $n$  строках заданы пары целых чисел  $x_i, y_i$  ( $1 \leq x_i, y_i \leq 10^6$ ) — координаты  $i$ -го участника олимпиады.

В следующих  $m$  строках заданы тройки целых положительных чисел  $X_i, Y_i, a_i$  ( $1 \leq X_i, Y_i \leq 10^6, 1 \leq a_i \leq n$ ) — координаты  $i$ -й площадки и ее вместимость.

Гарантируется, что сумма  $a_i$  не меньше  $n$ , а так же что она не превосходит 1000.

### Формат выходных данных

Выведите одно число — максимальное расстояние, которое придется пройти участнику до точки проведения олимпиады при оптимальном распределении. Ответ будет считаться верным, если его абсолютная или относительная ошибка не будет превосходить  $10^{-6}$ . Формально, пусть ваш ответ равен  $a$ , а ответ жюри —  $b$ . Ваш ответ считается правильным, если  $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
3 2 1 1 2 3 3 2 1 1 1 2 2 2	1.0000000000
3 2 100 100 101 101 102 102 101 101 2 105 105 2	4.2426406871

### Замечание

В первом примере надо отправить первого участника на первую площадку, а оставшихся двух — на вторую. Тогда первый участник пройдет расстояние 0, а второй и третий — по 1.

Во втором примере надо первого и второго участников отправить на площадку 1, а третьего — на площадку 2. Тогда первый участник пройдёт расстояние 1.41421356237, второй — 0, а третий — 4.2426406871.

## Система оценки

Тесты к этой задаче состоят из девяти 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	Дополнительные ограничения			Необх. группы	Комментарий
		$n$	$m$	$a_i$		
0	0	–	–	–	–	Тесты из условия.
1	10	–	–	$a_i = n$	–	
2	11	–	$m \leq 2$	–	0	
3	15	–	$m \leq 3$	–	0, 2	
4	14	$n \leq 10$	$m \leq 10$	–	0	
5	13	$n \leq 50$	$m \leq 50$	–	0, 4	
6	7	$n \leq 200$	$m \leq 200$	–	0, 4, 5	
7	9	$n \leq 300$	$m \leq 300$	–	0, 4, 5, 6	
8	10	$n \leq 400$	$m \leq 400$	–	0, 4, 5, 6, 7	<b>Offline-проверка.</b>
9	11	–	–	–	0–8	<b>Offline-проверка.</b>

## Задача G. Тир

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:	512 мегабайт

Мальчик Витя любит играть с огнестрельным оружием. Это увлечение и привело его в ближайший тир.

Если посмотреть на тир сверху, то он имеет форму прямоугольник размера  $n \times m$ , где в центре каждого единичного отрезка на верхней стороне прямоугольника, имеющей длину  $m$ , находится пистолет, а в центре каждого единичного отрезка на нижней стороне находится мишень. Каждый раз, когда Витя покупает жетон, он получает  $m$  пуль и стреляет из каждого пистолета ровно по одному разу. Витя — отличный стрелок, поэтому пуля после выстрела летит строго перпендикулярно верхней стороне. Поскольку между пистолетом и мишенью нет препятствий, пуля всегда попадает в мишень.

Вите надоело каждый раз попадать во все  $m$  мишеней, поэтому он решил усложнить задачу. Для этого он планирует поставить перегородки в некоторые клетки. Перегородки располагаются под углом  $45^\circ$  и соединяют один из углов клетки с противоположным. Когда летящая пуля ударяется об перегородку, она отражается в соответствии с законами физики, то есть поворачивает на  $90^\circ$  влево или вправо в зависимости от расположения перегородки. Пуля продолжает лететь до тех пор, пока не попадёт в мишень или не вылетит за пределы тира.

После выставления перегородок пули, выпущенные из некоторых пистолетов, могут не долетать до мишеней, поэтому Вите интересно, во сколько мишеней он попадёт хотя бы единожды.

Чтобы усложнить вам задачу, Витя будет добавлять и удалять перегородки и после каждого изменения спрашивать, во сколько мишеней он попадёт хотя бы один раз, если выстрелит сейчас из каждого пистолета.

Изначально в тире нет ни одной перегородки.

### Формат входных данных

В первой строке заданы три числа  $n, m, q$  ( $1 \leq n, m, q \leq 200\,000$ ) — высота тира, ширина тира и число запросов, соответственно.

В следующих  $q$  строках описываются изменения.

Каждое изменение задано тройкой чисел  $a_i, x_i$  и  $y_i$  ( $0 \leq a_i \leq 2, 1 \leq x_i \leq n, 1 \leq y_i \leq m$ ).

- Если  $a_i = 0$ , то в  $i$ -м запросе требуется удалить перегородку, находящуюся в  $x_i$ -й строке и  $y_i$ -м столбце. Гарантируется, что на момент  $i$ -го запроса она там есть.
- Если  $a_i = 1$ , то в  $i$ -м запросе требуется добавить перегородку из левого верхнего в правый нижний угол клетки, находящейся на пересечении  $x_i$ -й строки и  $y_i$ -го столбца. Гарантируется, что на момент  $i$ -го запроса там нет перегородки.
- Если  $a_i = 2$ , то в  $i$ -м запросе требуется добавить перегородку из левого нижнего в правый верхний угол клетки, находящейся на пересечении  $x_i$ -й строки и  $y_i$ -го столбца. Гарантируется, что на момент  $i$ -го запроса там нет перегородки.

Строки пронумерованы от 1 до  $n$  сверху вниз, столбцы пронумерованы от 1 до  $m$  слева направо.

### Формат выходных данных

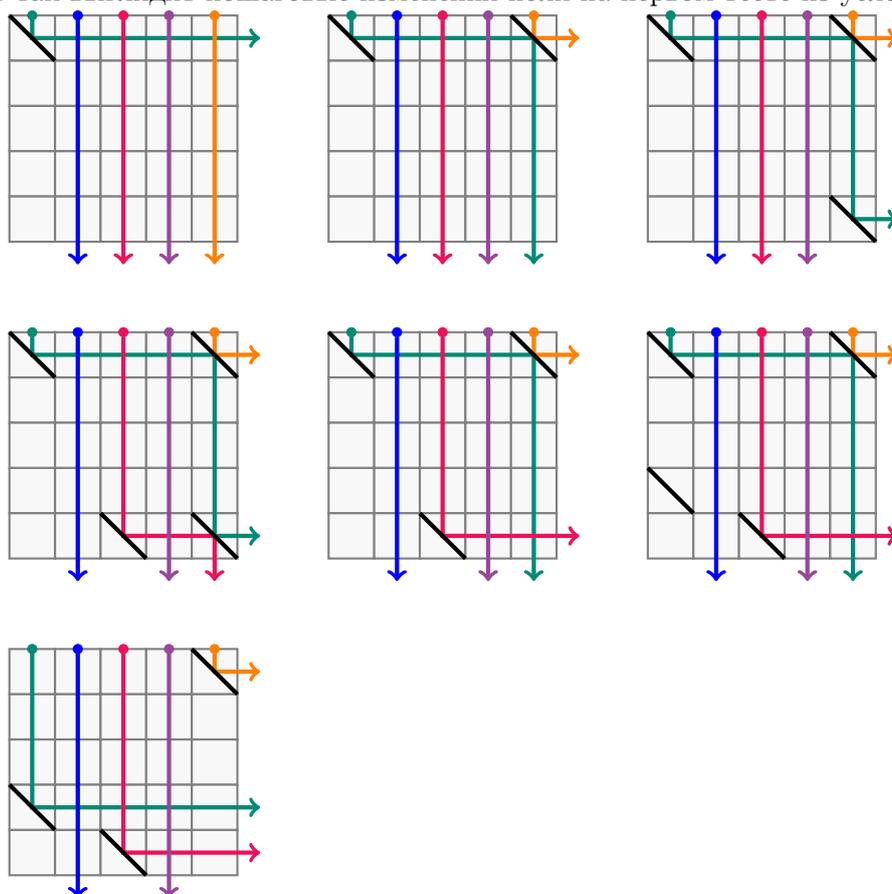
Выведите  $q$  чисел,  $i$ -е из них должно быть равно числу мишеней, в которые попадёт Витя после  $i$ -го изменения.

## Примеры

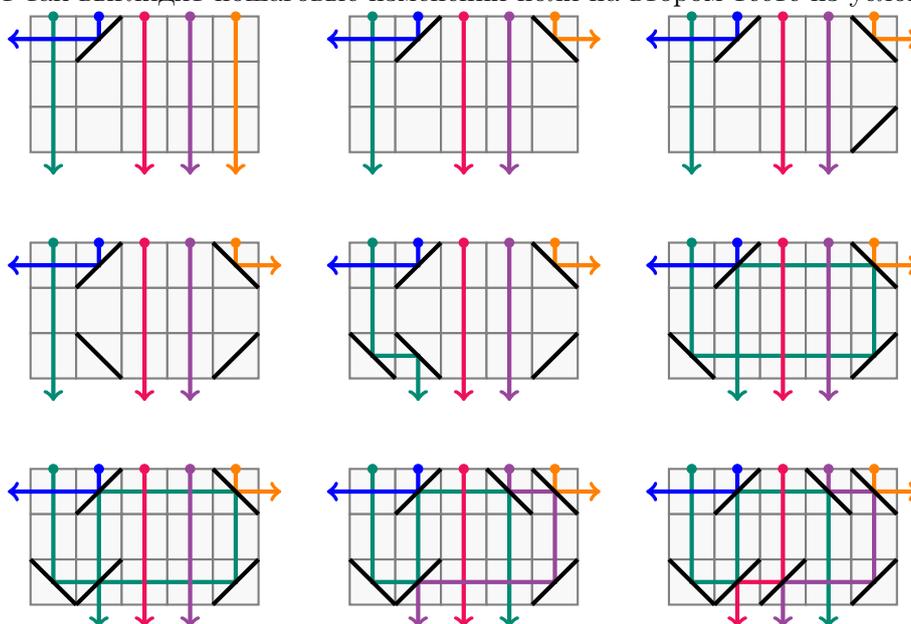
стандартный ввод	стандартный вывод
5 5 7 1 1 1 1 1 5 1 5 5 1 5 3 0 5 5 1 4 1 0 1 1	4 4 3 3 3 3 2
3 5 9 2 1 2 1 1 5 2 3 5 1 3 2 1 3 1 0 3 2 2 3 2 1 1 4 2 3 3	4 3 3 3 3 3 3 3 3
5 5 11 1 5 1 2 5 5 1 1 5 2 1 2 2 5 2 0 5 2 1 3 2 2 1 1 0 1 2 1 1 3 2 5 3	4 3 3 3 3 3 2 2 2 2 2

## Замечание

Вот так выглядят пошаговые изменения поля на первом тесте из условия:



Вот так выглядят пошаговые изменения поля на втором тесте из условия:



## Система оценки

Тесты к этой задаче состоят из восьми 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
		$n, m, q$	$a_i$		
0	0	-	-	-	Тесты из условия.
1	9	$n, m, q \leq 500$	$a_i \leq 1$		
2	11	$n, m, q \leq 500$	-	0, 1	
3	11	$n, m, q \leq 10\,000$	$a_i \leq 1$	1	
4	12	$n, m, q \leq 10\,000$	-	0–3	
5	13	$n, m, q \leq 60\,000$	$a_i \leq 1$	1, 3	
6	15	$n, m, q \leq 60\,000$	-	0–5	
7	13	-	$a_i \leq 1$	1, 3, 5	<b>Offline-проверка.</b>
8	16	-	-	0–7	<b>Offline-проверка.</b>

## Задача Н. Украшение дома

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:	512 мегабайт

Даня играет со своими друзьями в известную игру `minesart`. На своем сервере он построил дом, передняя стена которого состоит из  $n$  столбиков квадратных блоков,  $i$ -й столбик имеет высоту  $h_i$ .

Недавно в игре вышло обновление, в котором в игру были добавлены прямоугольные картины. Теперь Даня хочет выбрать некоторую картину и украсить картинами такого размера переднюю сторону дома так, чтобы каждый блок стены был покрыт хотя бы одной картиной. Картины можно поворачивать на 90 градусов, накладывать друг на друга, но нельзя, чтобы под каким-то блоком картины не было блока стены.

Даня считает, что красота картины равна её площади. Помогите Дане найти картину максимальной красоты, такую что её копиями и поворотами можно украсить всю переднюю стену дома.

### Формат входных данных

В первой строке содержится одно целое число  $n$  ( $1 \leq n \leq 200\,000$ ) — количество столбиков на передней стене дома.

Во второй строке содержатся  $n$  целых чисел  $h_1, h_2, \dots, h_n$  ( $0 \leq h_i \leq 10^9$ ) — высоты столбиков.

Гарантируется, что хотя бы один столбик имеет высоту больше 0.

### Формат выходных данных

Выведите максимальную площадь картины, такую что такими картинами и её поворотами Даня сможет украсить всю переднюю стену дома.

### Примеры

стандартный ввод	стандартный вывод
5 4 4 1 4 4	4
7 2 4 4 2 0 4 4	8
6 3 6 6 6 5 4	15

### Замечание

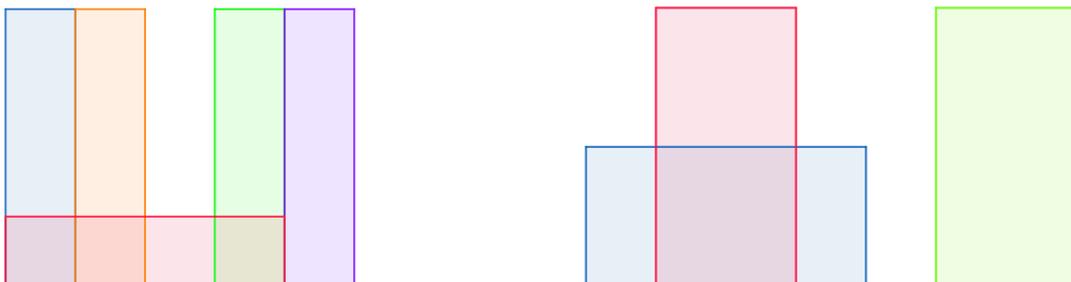
В первом примере подойдёт картина длиной 1 и высотой 4.

Во втором примере подойдёт картина длиной 2 и высотой 4.

В третьем примере подойдёт картина длиной 3 и высотой 5.

Можно показать, что нельзя покрыть заборы прямоугольниками большей площади.

На рисунках ниже показаны примеры покрытия картинами стены в первом и втором примерах:



## 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
		$n$	$h_i$		
0	0	–	–	–	Тесты из условия.
1	11	$n \leq 10$	$h_i \leq 10$	0	
2	12	$n \leq 500$	$h_i \leq 500$	0, 1	
3	18	$n \leq 5000$	$h_i \leq 5000$	0, 1, 2	
4	16	$n \leq 5000$	–	0–3	
5	18	–	$h_i \leq 500$	0, 1, 2	
6	25	–	–	0–5	

## Задача I. Доставка посылок

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	512 мегабайт

В городе Нске есть  $n$  домов, которые соединены  $n - 1$  двусторонней дорогой так, что от любого дома можно доехать до любого другого, передвигаясь по дорогам. В каждом доме проживает ровно один человек.

В один праздничный день в качестве подарка каждый житель города решил отправить каждому посылку. Посылки в Нске доставляет специальная бесконтактная почта. Чтобы доставить жителю дома  $u$  посылку от жителя дома  $v$ , почтовый курьер забирает посылку около дома  $v$ , затем едет по кратчайшему пути до дома  $u$  и оставляет посылку около него.

К сожалению, в Нске действуют строгие правила отправки посылок. Вес каждой посылки должен быть целочисленным, а также для каждой дороги установлено целое число  $w_i$  — ограничение на провоз посылок, означающее, что по этой дороге можно провозить только посылки, вес которых делит  $w_i$ . Иными словами, посылка веса  $x$  может быть провезена по дороге с ограничением  $w_i$  только если  $w_i$  делится нацело на  $x$ .

Жители города очень любят праздники, поэтому каждый житель решил отправить каждому посылку наибольшего возможного веса.

Чтобы обеспечить наилучшую работу бесконтактной почты, необходимо посчитать суммарный вес всех отправленных посылок. Помогите решить эту важную задачу!

### Формат входных данных

Первая строка содержит целое число  $n$  ( $2 \leq n \leq 100\,000$ ) — количество домов в городе.

Каждая из следующих  $n - 1$  строк содержит описание дороги: три целых числа  $u_i, v_i$  и  $w_i$  ( $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^6$ ) — номера домов, между которыми проложена  $i$ -я дорога, и ограничение на провоз посылок в  $i$ -й дороге. Дома в городе нумеруются, начиная с единицы.

Гарантируется, что между каждой парой домов существует путь по дорогам.

### Формат выходных данных

Выведите единственное число — сумму весов всех отправленных жителями посылок.

### Примеры

стандартный ввод	стандартный вывод
3 1 2 6 1 3 4	24
6 1 2 4 3 4 3 1 4 10 5 1 2 4 6 5	82

### Замечание

В первом примере из условия житель 1 мог отправить жителю 2 посылку веса не более 6, аналогично житель 2 мог отправить жителю 1 посылку веса не более 6. Житель 1 мог отправить жителю 3 посылку веса не более 4, аналогично житель 3 мог отправить жителю 1 посылку веса не более 4. Житель 2 мог отправить жителю 3 посылку веса не более 2, так как число 6 должно делиться на вес посылки, и число 4 должно делиться на вес посылки. Аналогично житель 3 мог отправить жителю 2 посылку веса не более 2.

В таблице ниже отмечены максимальные веса, которые могут отправить друг другу жители во втором примере из условия.

Житель	1	2	3	4	5	6
1	–	4	1	10	2	5
2	4	–	1	2	2	1
3	1	1	–	3	1	1
4	10	2	3	–	2	5
5	2	2	1	2	–	1
6	5	1	1	5	1	–

## Система оценки

Тесты к этой задаче состоят из восьми групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** 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
		$n$	$w_i$		
0	0	–	–	–	Тесты из условия.
1	14	$n \leq 500$	$w_i \leq 100\,000$	0	
2	11	$n \leq 3000$	–	0, 1	
3	12	–	–	–	Все $w_i$ простые числа.*
4	15	–	–	–	$u_i = 1, v_i = i + 1$
5	16	–	–	–	$u_i = i, v_i = i + 1$
6	10	–	$w_i \leq 500$	0	
7	8	–	$w_i \leq 10\,000$	0, 6	<b>Offline-проверка.</b>
8	14	–	–	0–7	<b>Offline-проверка.</b>

\*Простым числом называется число, имеющее ровно два различных натуральных делителя — единицу и самого себя.

## Задача J. Тараканы общежития

Имя входного файла:	стандартный ввод или <code>input.txt</code>
Имя выходного файла:	стандартный вывод или <code>output.txt</code>
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Виталий — первокурсник престижного университета. Из-за коронавируса и новых ограничительных мер большинство студентов уезжают из общежития, но Виталий наоборот решил заселиться. К его удивлению, помимо немногочисленных однокурсников, в общежитии его ожидало полчище тараканов.

После непродолжительного подсчёта было выявлено, что всего в комнате Виталия без спроса проживают  $n$  тараканов. Также было замечено, что живут они лишь на полу. Для простоты скажем, что они представляют из себя целочисленные точки на плоскости. Тараканы крайне ленивые, поэтому они совсем не двигаются.

Виталий не переносит насекомых, тем более тараканов, поэтому он решил от них избавиться. Так как распаковать вещи времени особо не было, самым действенным подручным орудием оказался простой тапок. После некоторого числа тестов было выявлено, что если бить плоской стороной, тараканы спокойно переживают удар. Таким образом, надо бить боковой стороной, но в таком случае зона поражения представляет собой линию, то есть погибают только тараканы, находящиеся на этой прямой.

Виталий готов сделать только  $k$  ударов и если после этого получится так, что остался хотя бы один живой таракан, то студент в ужасе убежит. Помогите Виталию определить как ему следует бить.

### Формат входных данных

В первой строке вводится  $t$  ( $1 \leq t \leq 50$ ) — количество наборов входных данных.

Описание каждого набора входных данных начинается с двух целых чисел  $n$  и  $k$  ( $1 \leq n \leq 100\,000$ ,  $1 \leq k \leq 12$ ) — количество тараканов на полу и число ударов, которое готов сделать Виталий, соответственно.

В следующих  $n$  строках вводится по два целых числа  $x_i, y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) — координаты  $i$ -го таракана. Гарантируется, что координаты всех тараканов различны.

Наборы входных данных отделены пустой строкой.

Гарантируется, что сумма  $n$  в наборах входных данных не превосходит 100 000.

Также гарантируется, что тест содержит не более одного набора входных данных с  $k \geq 11$ .

### Формат выходных данных

Для каждого набора входных данных выведите «No» (без кавычек), если нельзя убить тараканов за  $k$  ударов.

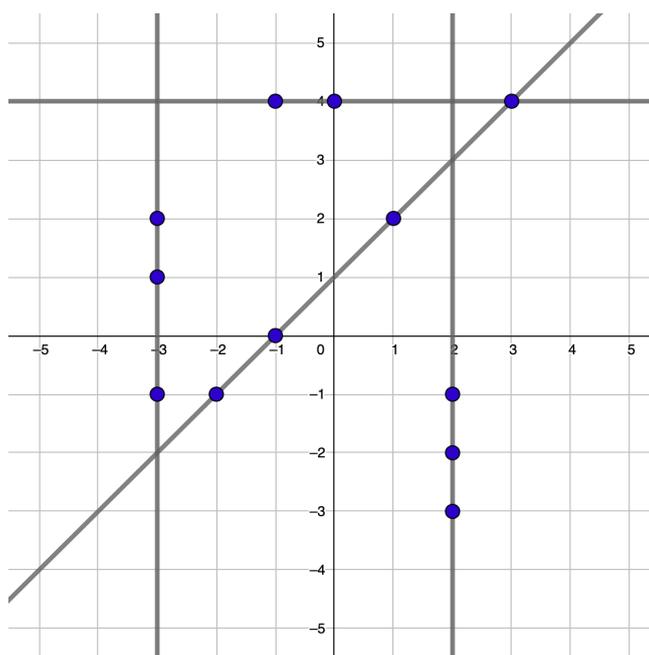
Иначе выведите «Yes», а в следующей строке целое число  $m$  ( $1 \leq m \leq k$ ) — количество ударов, которое необходимо сделать Виталию. В следующих  $m$  строках выведите по четыре целых числа  $x_{i,1}, y_{i,1}, x_{i,2}, y_{i,2}$  ( $|x_{i,1}|, |y_{i,1}|, |x_{i,2}|, |y_{i,2}| \leq 10^9$ ) — координаты двух точек, лежащих на прямой  $i$ -го удара.

## Пример

стандартный ввод	стандартный вывод
2	Yes
12 4	4
1 2	-2 -1 -1 0
-3 1	-3 -1 -3 1
2 -2	2 -3 2 -2
2 -1	-1 4 0 4
-1 0	No
2 -3	
-1 4	
-2 -1	
0 4	
-3 2	
-3 -1	
3 4	
12 3	
1 2	
-3 1	
2 -2	
2 -1	
-1 0	
2 -3	
-1 4	
-2 -1	
0 4	
-3 2	
-3 -1	
3 4	

## Замечание

Рисунок ниже иллюстрирует первый набор входных данных в тесте из условия:



## Система оценки

Тесты к этой задаче состоят из четырнадцати групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **необходимых** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Обратите внимание, дополнительные ограничения на  $n$  не ограничивают сумму  $n$  по всем наборам входных данных.

Группа	Баллы	Дополнительные ограничения			Необх. группы	Комментарий
		$n$	$k$	$t$		
0	0	–	–	–	–	Тесты из условия.
1	3	$n \leq 100$	$k \leq 1$	–	–	
2	3	–	$k \leq 1$	–	1	
3	6	$n \leq 100$	$k \leq 2$	–	1	
4	6	–	$k \leq 2$	–	1–3	
5	6	$n \leq 100$	$k \leq 4$	–	0, 1, 3	
6	6	–	$k \leq 4$	–	0–5	
7	10	$n \leq 100$	$k \leq 6$	–	0, 1, 3, 5	
8	10	–	$k \leq 6$	–	0–7	
9	9	$n \leq 100$	$k \leq 8$	–	0, 1, 3, 5, 7	
10	9	–	$k \leq 8$	–	0–9	
11	6	$n \leq 100$	$k \leq 10$	–	0, 1, 3, 5, 7, 9	<b>Offline-проверка.</b>
12	6	–	$k \leq 10$	–	0–11	<b>Offline-проверка.</b>
13	10	$n \leq 100$	–	$t = 1$	0, 1, 3, 5, 7, 9, 11	<b>Offline-проверка.</b>
14	10	–	–	$t = 1$	0–13	<b>Offline-проверка.</b>