

Задача 5. Удаление чисел

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

В ряд выписаны натуральные числа от 1 до n и задано натуральное число k .

Выполняется один или несколько шагов по удалению чисел в этом ряду. На очередном шаге оставшиеся числа просматриваются в возрастающем порядке, и каждое k -е число удаляется. Если после очередного шага осталось меньше k чисел, то процесс удаления чисел завершается.

Необходимо определить, на каком шаге будет удалено число n , или выяснить, что оно не будет удалено до завершения процесса.

Например, пусть $n = 13$, $k = 2$.

- На первом шаге будут удалены числа 2, 4, 6, 8, 10 и 12, останутся числа 1, 3, 5, 7, 9, 11 и 13.
- На втором шаге будут удалены числа 3, 7 и 11, останутся числа 1, 5, 9 и 13.
- На третьем шаге будут удалены числа 5 и 13, останутся числа 1 и 9.
- На четвертом шаге будет удалено число 9, останется число 1. Поскольку осталось одно число, процесс завершается.

Таким образом, число 13 будет удалено на третьем шаге.

Требуется написать программу, которая по заданным числам n и k определяет, на каком шаге будет удалено число n .

Формат входных данных

Первая строка входных данных содержит целое число n ($3 \leq n \leq 10^{18}$).

Вторая строка входных данных содержит целое число k ($2 \leq k \leq 100$, $k < n$).

Формат выходных данных

Требуется вывести одно целое число — номер шага, на котором будет удалено число n , или число 0, если число n не будет удалено.

Пример входных и выходных данных

входные данные	выходные данные
13 2	3

Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи	Информация о проверке
		n	k		
1	16	$3 \leq n \leq 1000$	$k = 2$		полная
2	10	$3 \leq n \leq 10^{18}$	$k = 2$	1	полная
3	14	$3 \leq n \leq 1000$	$2 \leq k \leq 100$, $k < n$	1	полная
4	20	$3 \leq n \leq 10^6$	$2 \leq k \leq 100$, $k < n$	1, 3	полная
5	40	$3 \leq n \leq 10^{18}$	$2 \leq k \leq 100$, $k < n$	1 – 4	полная

Задача 6. Старая книга

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

Группа юных археологов работает на раскопках здания древней библиотеки. Летом они обнаружили остатки старой книги и, изучив их, сделали следующие выводы.

Книга содержит несколько страниц, каждая страница содержит либо текст, либо иллюстрацию. Первые k страниц книги точно содержат иллюстрации. Все страницы книги пронумерованы, но номер страницы написан только на страницах, содержащих текст. Сумма номеров страниц с текстом равна s .

К сожалению, ни общее количество страниц в книге, ни количество иллюстраций установить не удалось. Тем не менее, юных археологов заинтересовал вопрос, какое минимальное количество иллюстраций могло быть в книге.

Например, если $k = 1$, а $s = 8$, то страницы книги могли иметь следующее содержание (буквой «Т» обозначена страница, содержащая текст, а буквой «И» — страница, содержащая иллюстрацию):

- И Т И И И Т, пронумерованы страницы 2 и 6, 4 иллюстрации;
- И И Т И Т, пронумерованы страницы 3 и 5, 3 иллюстрации;
- И И И И И И И Т, пронумерована страница 8, 7 иллюстраций.

Минимальное количество иллюстраций равно 3.

Требуется написать программу, которая по заданным целым числам k и s определяет минимальное количество иллюстраций, которое могло быть в книге.

Формат входных данных

Первая строка входных данных содержит целое число k ($0 \leq k \leq 10^9$).

Вторая строка входных данных содержит целое число s ($k + 1 \leq s \leq 10^{12}$).

Формат выходных данных

Требуется вывести одно целое число — минимальное количество иллюстраций в книге.

Пример входных и выходных данных

входные данные	выходные данные
1 8	3

Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи	Информация о проверке
		k	s		
1	15	$k = 0$	$1 \leq s \leq 200$		полная
2	20	$k = 0$	$1 \leq s \leq 10^{12}$	1	полная
3	30	$0 \leq k \leq 199$	$k + 1 \leq s \leq 200$	1	полная
4	35	$0 \leq k \leq 10^9$	$k + 1 \leq s \leq 10^{12}$	1, 2, 3	полная

Задача 7. Красота фейерверка

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

В лаборатории теоретической пиротехники изучают новые технологии организации фейерверков. Фейерверк представляется как корневое дерево, а поскольку в мощном фейерверке его элементы также взрываются, порождая новые фейерверки, то ученые вводят операцию возведения корневого дерева в степень.

Корневое дерево содержит одну или несколько вершин. Одна из вершин выделена и называется *корнем* дерева, для каждой из остальных вершин ровно одна другая вершина является *родителем*. При этом от любой вершины можно добраться до корня, последовательно переходя от вершины к ее родителю. Вершина, которая не является родителем никакой другой вершины, называется *листом*. Если вершина x является родителем вершины y , то вершина y является *ребенком* вершины x . Будем говорить, что вершина и ее родитель *соединены ребром*.

На рис. 1 показан пример корневого дерева с корнем в вершине 1. Родителем вершин 2 и 3 является вершина 1, родителем вершины 4 является вершина 2. Вершины 2 и 3 — дети вершины 1, а вершина 4 — ребенок вершины 2. Листьями являются вершины 3 и 4.

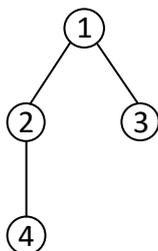


Рис. 1. Пример корневого дерева с корнем в вершине 1, листьями 3 и 4.

Фейерверк задается своим *базовым деревом* T и *мощностью* m . Фейерверк представляется деревом, которое получается в результате возведения дерева T в степень m . Операция возведения дерева в степень устроена следующим образом. Если $m = 1$, то результат T^1 — само дерево T . Для $m > 1$ рассмотрим дерево T^{m-1} . Выполним следующую операцию: для каждого листа x дерева T^{m-1} создадим копию дерева T и назначим лист x родителем корня соответствующей копии. Получившееся дерево будет деревом T^m .

На рис. 2 показано дерево, представленное на рис. 1, в степенях 1, 2 и 3.

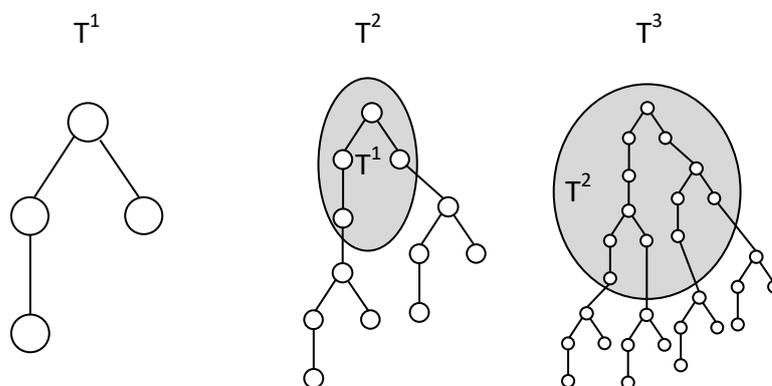


Рис. 2. Пример возведения дерева в степени 1, 2 и 3.

Путь в дереве называется последовательность вершин, в которой две соседние вершины соединены ребром. Все вершины в пути должны быть различны.

Для того, чтобы оценить красоту фейерверка, необходимо определить, какое максимальное количество вершин может содержать путь в дереве, которым представляется фейерверк. На рис. 3 приведен путь в дереве T^2 , содержащий максимальное количество вершин. Таким образом, красота фейерверка с базовым деревом T и мощностью 2 равна 10.

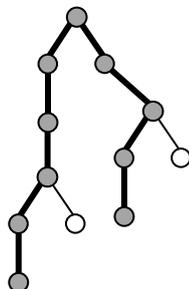


Рис. 3. Путь в дереве T^2 , содержащий максимальное количество вершин.

Требуется написать программу, которая по описанию дерева T и натуральному числу m определяет красоту фейерверка с базовым деревом T и мощностью m .

Формат входных данных

Первая строка входных данных содержит два натуральных числа n и m — количество вершин в базовом дереве фейерверка T и его мощность ($3 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$).

Вторая строка описывает дерево T и содержит $(n - 1)$ целых чисел: p_2, p_3, \dots, p_n — номера родителей вершин $2, 3, \dots, n$, соответственно ($1 \leq p_i \leq i - 1$).

Формат выходных данных

Требуется вывести одно целое число — красоту фейерверка, представляемого деревом T^m .

Пример входных и выходных данных

входные данные	выходные данные
4 2 1 1 2	10

Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения		Необходимые подзадачи	Информация о проверке
		n	m		
1	19	$3 \leq n \leq 5000$	$m = 1$		полная
2	10	$3 \leq n \leq 200\,000$	$m = 1$	1	полная
3	20	$3 \leq n \leq 5000$	$1 \leq m \leq 5000$	1	полная
4	19	$3 \leq n \leq 5000$	$1 \leq m \leq 200\,000$	1, 3	полная
5	32	$3 \leq n \leq 200\,000$	$1 \leq m \leq 200\,000$	1 – 4	полная

Задача 8. Обработка больших данных

Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

В научной лаборатории разрабатывается новая архитектура суперкомпьютера, позволяющая эффективно обрабатывать большие объемы данных.

Суперкомпьютер имеет 2^k ячеек памяти, пронумерованных от 0 до $2^k - 1$. *Отрезком* $[L, R]$ называется последовательность подряд идущих ячеек памяти с номерами от L до R , включительно.

Некоторые отрезки памяти являются *корректными*. Отрезок памяти $[L, R]$ является корректным, если его длина $(R - L + 1)$ равна 2^i для некоторого i , а число L делится на 2^i . Например, если $k = 3$, то ячейки памяти пронумерованы от 0 до 7, а корректными являются отрезки $[0, 7]$, $[0, 3]$, $[4, 7]$, $[0, 1]$, $[2, 3]$, $[4, 5]$, $[6, 7]$, $[0, 0]$, $[1, 1]$, $[2, 2]$, $[3, 3]$, $[4, 4]$, $[5, 5]$, $[6, 6]$ и $[7, 7]$.

Ключевой операцией в новой архитектуре является операция STORE, которая за одно действие присваивает одно и то же значение всем ячейкам памяти некоторого *корректного* отрезка.

Исходно все ячейки памяти содержат значение 0. В лаборатории планируют запустить на суперкомпьютере программу обработки данных, но перед её запуском необходимо инициализировать память определенным образом. А именно: первые c_1 ячеек памяти необходимо заполнить значениями v_1 , следующие c_2 ячеек — значениями v_2 , и так далее, последние c_n ячеек памяти необходимо заполнить значениями v_n , где $1 \leq v_i \leq m$.

Ученым надо выяснить, какое минимальное количество операций STORE необходимо выполнить, чтобы проинициализировать память требуемым образом.

Например, если $k = 3$, $n = 3$, $m = 2$, $c_1 = 1$, $v_1 = 1$, $c_2 = 2$, $v_2 = 2$, $c_3 = 5$, $v_3 = 1$, то итоговое содержимое памяти должно быть следующим: $[1, 2, 2, 1, 1, 1, 1, 1]$. В этом случае для инициализации памяти достаточно трех операций STORE:

- STORE($[0, 7], 1$), после этой операции все ячейки памяти содержат значение 1;
- STORE($[1, 1], 2$), после этой операции содержимое памяти $[1, 2, 1, 1, 1, 1, 1, 1]$;
- STORE($[2, 2], 2$), после этой операции содержимое памяти $[1, 2, 2, 1, 1, 1, 1, 1]$, как и требуется.

Заметим, что операцию STORE($[1, 2], 2$) выполнить нельзя, потому что $[1, 2]$ не является корректным отрезком памяти.

Требуется написать программу, которая по заданному содержимому памяти определяет минимальное количество операций STORE, которое необходимо выполнить для инициализации памяти заданным образом.

Формат входных данных

Первая строка входных данных содержит три целых числа: k , n и m ($0 \leq k \leq 30$, $1 \leq n \leq 10^5$, $1 \leq m \leq 10^9$).

Следующие n строк содержат по два целых числа, i -я из этих строк содержит числа c_i и v_i ($1 \leq c_i \leq 2^k$, $1 \leq v_i \leq m$, $c_1 + c_2 + \dots + c_n = 2^k$).

Формат выходных данных

Требуется вывести одно целое число — минимальное количество операций STORE, которое необходимо выполнить для инициализации памяти заданным образом.

Примеры входных и выходных данных

входные данные	выходные данные
3 3 2 1 1 2 2 5 1	3

Описание подзадач и системы оценивания

Баллы за каждую подзадачу начисляются только в случае, если все тесты для этой подзадачи и необходимых подзадач успешно пройдены.

Подзадача	Баллы	Дополнительные ограничения			Необходимые подзадачи	Информация о проверке
		k	n	m		
1	15	$0 \leq k \leq 3$	$1 \leq n \leq 8$	$1 \leq m \leq 8$		баллы
2	15	$0 \leq k \leq 19$	$1 \leq n \leq 10^5$	$1 \leq m \leq 10$	1	баллы
3	15	$0 \leq k \leq 30$	$1 \leq n \leq 10^5$	$1 \leq m \leq 10$	1, 2	баллы
4	10	$0 \leq k \leq 30$	$1 \leq n \leq 10^5$	$1 \leq m \leq 50$	1, 2, 3	баллы
5	15	$0 \leq k \leq 19$	$1 \leq n \leq 10^5$	$1 \leq m \leq 10^9$	1, 2	баллы
6	30	$0 \leq k \leq 30$	$1 \leq n \leq 10^5$	$1 \leq m \leq 10^9$	1 – 5	баллы