

Задача 1. Улучшение успеваемости

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

В лицее на уроках информатики ответы учеников оцениваются целым числом баллов от 2 до 5. Итоговая оценка по информатике выставляется как среднее арифметическое оценок на всех уроках, округленное до ближайшего целого числа. Если среднее значение находится ровно посередине между двумя целыми числами, то оценка округляется вверх.

Примеры округления оценок приведены в таблице.

Оценки на уроках	Среднее арифметическое	Итоговая оценка
2, 3, 5	$\frac{2 + 3 + 5}{3} = 3\frac{1}{3}$	3
3, 3, 4, 4	$\frac{3 + 3 + 4 + 4}{4} = 3\frac{1}{2}$	4
5, 5, 5, 3, 5	$\frac{5 + 5 + 5 + 3 + 5}{5} = 4\frac{3}{5}$	5

Все ученики лицея стремятся получить итоговую оценку по информатике не ниже 4 баллов. К сожалению, один из учеников получил на уроках a двоек, b троек и c четверок. Теперь он планирует получить несколько пятерок, причем хочет, чтобы итоговая оценка была не меньше 4 баллов. Ему надо понять, какое минимальное количество пятерок ему необходимо получить, чтобы добиться своей цели.

Требуется написать программу, которая по заданным целым неотрицательным числам a , b и c определяет минимальное количество пятерок, которое необходимо получить ученику, чтобы его итоговая оценка по информатике была не меньше 4 баллов.

Формат входных данных

Входные данные содержат три строки. Первая строка содержит целое неотрицательное число a , вторая строка содержит целое неотрицательное число b , третья строка содержит целое неотрицательное число c ($0 \leq a, b, c \leq 10^{15}$, $a + b + c \geq 1$).

Формат выходных данных

Выходные данные должны содержать одно число: минимальное число пятерок, которое необходимо получить ученику, чтобы итоговая оценка была не меньше 4 баллов.

Примечание

Следует обратить внимание, что входные данные в этой и других задачах не помещаются в стандартный 32-битный тип данных. Необходимо использовать 64-битный тип данных (`long long` в C++, `int64` в Паскале, `long` в Java).

Пример входных и выходных данных

входные данные	выходные данные
2 0 0	2

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
1	13	$1 \leq a \leq 100, b = 0, c = 0$ (ученик получал только двойки)		полная
2	14	$a = 0, 1 \leq b \leq 100, c = 0$ (ученик получал только тройки)		полная
3	15	$0 \leq a, b, c \leq 100$	1, 2	полная
4	28	$0 \leq a, b, c \leq 10^6$	1, 2, 3	полная
5	30	$0 \leq a, b, c \leq 10^{15}$	1, 2, 3, 4	полная

Задача 2. Квадраты и кубы

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

В лаборатории теории чисел одного университета изучают связь между распределением квадратов и кубов натуральных чисел.

Пусть задано целое неотрицательное число k . Рассмотрим множество натуральных чисел от a до b , включительно. Будем называть k -плотностью этого множества количество пар натуральных чисел x и y , таких, что $a \leq x^2 \leq b$, $a \leq y^3 \leq b$, причем $|x^2 - y^3| \leq k$.

Например, 2-плотность множества натуральных чисел от 1 до 30 равна 3, так как подходят следующие пары:

- $x = 1, y = 1, |x^2 - y^3| = |1 - 1| = 0$;
- $x = 3, y = 2, |x^2 - y^3| = |9 - 8| = 1$;
- $x = 5, y = 3, |x^2 - y^3| = |25 - 27| = 2$.

Требуется написать программу, которая по заданным натуральным числам a и b , а также целому неотрицательному числу k , определяет k -плотность множества натуральных чисел от a до b , включительно.

Формат входных данных

Входные данные содержат три строки. Первая строка содержит натуральное число a , вторая строка содержит натуральное число b , третья строка содержит целое неотрицательное число k ($1 \leq a \leq b \leq 10^{18}$, $0 \leq k \leq 10^{18}$).

Формат выходных данных

Выходные данные должны содержать одно целое число: искомую k -плотность множества натуральных чисел от a до b , включительно.

Пример входных и выходных данных

входные данные	выходные данные
1 30 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
		a, b	k		
1	10	$1 \leq a \leq b \leq 1000$	$k = 0$		полная
2	10	$1 \leq a \leq b \leq 10^{18}$	$k = 0$	1	полная
3	15	$1 \leq a \leq b \leq 1000$	$0 \leq k \leq 10$	1	полная
4	15	$1 \leq a \leq b \leq 10^6$	$0 \leq k \leq 10$	1, 3	полная
5	15	$1 \leq a \leq b \leq 10^9$	$0 \leq k \leq 10$	1, 3, 4	полная
6	15	$1 \leq a \leq b \leq 10^9$	$0 \leq k \leq 10^9$	1, 3, 4, 5	полная
7	20	$1 \leq a \leq b \leq 10^{18}$	$0 \leq k \leq 10^{18}$	1 – 6	полная

Задача 3. Лифт

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

В современном многоэтажном офисе крупной компании установлен новый лифт. В компании работает n сотрудников. Для проверки эффективности системы управления лифтом требуется провести моделирование его работы в конце рабочего дня, когда все сотрудники должны покинуть здание и спуститься на первый этаж.

В здании m этажей, пронумерованных от 1 до m снизу-вверх. Известно, что i -й сотрудник подходит к лифту в секунду t_i на этаже a_i , чтобы спуститься на первый этаж.

На каждом этаже могут находиться люди, ожидающие лифт. Когда очередной сотрудник подходит к лифту, он вызывает лифт, если на этом этаже лифт еще не вызван, либо присоединяется к ожидающим лифт. Таким образом, помимо вызвавшего лифт, вместе с ним лифт могут ожидать и другие сотрудники.

В каждый момент времени не более одного вызова является *активным*.

Изначально лифт свободен и находится на первом этаже. Когда поступает первый вызов, этот вызов становится активным и лифт отправляется на соответствующий этаж. Если несколько вызовов поступает одновременно, активным становится вызов от сотрудника с меньшим номером.

Лифт перемещается между этажами со скоростью один этаж в секунду. Когда лифт оказывается на этаже, откуда был сделан активный вызов, в него заходят все, кто уже ожидает лифт на этом этаже, и лифт отправляется вниз на первый этаж, со скоростью один этаж в секунду.

При движении вниз лифт останавливается на тех этажах, в которых был сделан вызов на момент проезда лифта мимо этого этажа. Все ожидающие лифт сотрудники заходят в него и вызов на этом этаже сбрасывается. Когда лифт завершает движение на первом этаже, все люди выходят из лифта, а лифт ожидает следующего вызова.

Если в момент, когда лифт освободился, есть хотя бы один необслуженный вызов, активируется вызов, который поступил раньше других. Если несколько вызовов поступило одновременно, активируется вызов от сотрудника с меньшим номером. Лифт продолжает обслуживание описанным образом, пока все n сотрудников не окажутся на первом этаже.

Будем считать, что люди входят и выходят из лифта мгновенно. Каждую секунду сначала люди подходят и вызывают лифт, а затем выполняются соответствующие действия (лифт перемещается на соседний этаж, в него входят или из него выходят люди, принимается решение, на какой вызов лифт должен отреагировать).

Требуется написать программу, которая по описанию вызовов лифта для каждого сотрудника определяет, в какой момент этот сотрудник окажется на первом этаже.

Формат входных данных

Первая строка входных данных содержит целые числа n и m — количество людей, вызывающих лифт, и количество этажей в здании ($1 \leq n \leq 10^5$, $2 \leq m \leq 10^9$).

Следующие n строк описывают сотрудников, i -я из этих строк содержит два целых числа t_i и a_i — секунду, в которую i -й сотрудник подходит к лифту, и номер этажа, на котором это происходит ($1 \leq t_1 \leq t_2 \leq \dots \leq t_n \leq 10^9$, $2 \leq a_i \leq m$).

Формат выходных данных

Выходные данные должны содержать n целых чисел, для каждого сотрудника требуется вывести секунду, в которую он выйдет из лифта на первом этаже.

Пример входных и выходных данных

входные данные	выходные данные
5 4	6
2 3	12
2 4	6
5 2	12
5 3	12
9 3	

Пояснение к примеру

Пример работы лифта по шагам показан в следующей таблице.

Время в секундах	1 этаж	2 этаж	3 этаж	4 этаж
0	[]			
1	[]			
2	[]↑ ³		☺ ₁	☺ ₂
3		[]↑ ³	☺ ₁	☺ ₂
4			[]←☺ ₁	☺ ₂
5		[☺ ₁]←☺ ₃	☺ ₄	☺ ₂
6	[☺ ₁ →, ☺ ₃ →]↑ ⁴		☺ ₄	☺ ₂
7		[]↑ ⁴	☺ ₄	☺ ₂
8			[]↑ ⁴ ☺ ₄	☺ ₂
9			☺ ₄ , ☺ ₅	[]←☺ ₂
10			[☺ ₂]←☺ ₄ , ←☺ ₅	
11		[☺ ₂ , ☺ ₄ , ☺ ₅]		
12	[☺ ₂ →, ☺ ₄ →, ☺ ₅ →]			

Использованные в пояснении к примеру обозначения

Обозначение	Пояснение
[]	обозначение лифта
[]↑ ^j	лифт движется на активный вызов, сделанный на j-м этаже
☺ _i	i-й сотрудник ждет лифта
←☺ _i	i-й сотрудник заходит в лифт
[☺ _i]	i-й сотрудник находится в лифте
[☺ _i →]	i-й сотрудник выходит из лифта

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	t_i		
1	15	$n = 1$	$2 \leq m \leq 100$	$1 \leq t_i \leq 100$		полная
2	30	$1 \leq n \leq 100$	$2 \leq m \leq 100$	$1 \leq t_i \leq 100$	1	полная
3	16	$1 \leq n \leq 100$	$2 \leq m \leq 10^9$	$1 \leq t_i \leq 10^9$	1, 2	полная
4	12	$1 \leq n \leq 10^5$	$2 \leq m \leq 10^4$	$1 \leq t_i \leq 10^4$	1, 2	полная
5	27	$1 \leq n \leq 10^5$	$2 \leq m \leq 10^9$	$1 \leq t_i \leq 10^9$	1, 2, 3, 4	полная

Задача 4. Мониторинг труб

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

Газораспределительная система одного региона устроена следующим образом. Она содержит n узлов, пронумерованных от 1 до n , некоторые узлы соединены односторонними трубами. Узел с номером 1 соответствует центральному газохранилищу.

Система узлов описывается числами от p_2, p_3, \dots, p_n . Для всех i от 2 до n узел с номером p_i соединен односторонней трубой с узлом i , газ по этой трубе передается от узла p_i к узлу i . Известно, что возможно доставить газ по трубам от центрального газохранилища до любого узла системы (возможно, с использованием промежуточных узлов). В системе используются трубы различных типов, тип трубы обозначается буквой английского алфавита от «a» до «z». Труба, соединяющая узел p_i с узлом i , имеет тип c_i .

Для проверки качества труб используется специальный робот. Он помещается в систему труб в одном из узлов и перемещается по трубам, каждый раз проверяя трубу, по которой он перемещается. Робот может перемещаться по трубам только в том же направлении, в котором по трубе передается газ. Совершив одно или несколько перемещений по трубам между узлами, робот извлекается из системы труб.

Каждый запуск робота должен соответствовать одной из m заданных *спецификаций*, пронумерованных от 1 до m . Спецификация с номером t представляет собой строку s_t , состоящую из строчных букв английского алфавита. Запуск соответствует спецификации s_t , если количество перемещений робота по трубам во время запуска совпадает с длиной s_t , и для всех j от 1 до длины s_t на j -м шаге робот перемещается по трубе, тип которой совпадает с $s_t[j]$ — символом на позиции j в спецификации.

Если запуск робота соответствует спецификации с номером t , то стоимость этого запуска составляет w_t . Оператору системы необходимо проверить все трубы, для этого можно запускать робот несколько раз. Каждый раз выбирается спецификация и маршрут робота по трубам, соответствующие выбранной спецификации. Необходимо проверить все трубы так, чтобы суммарная стоимость запусков робота для проверки качества труб была минимальна. Одну и ту же трубу можно проверять несколько раз.

Требуется написать программу, которая по описанию системы труб и списку спецификаций определяет минимальную суммарную стоимость запусков робота, в результате которых все трубы будут проверены, а также список необходимых для этого запусков (по требованию).

Формат входных данных

В первой строке входных данных находятся три целых числа n, m и t — количество узлов системы труб, количество спецификаций запусков робота и параметр, указывающий, требуется ли вывести список запусков робота или только их минимальную суммарную стоимость ($1 \leq n \leq 500, 1 \leq m \leq 10^5, t$ равно 0 или 1).

В последующих $(n - 1)$ строках содержится информация о трубах, $(i - 1)$ -я из этих строк содержит разделенные пробелом значения p_i и c_i , где p_i — целое число, задающее номер узла, из которого ведет труба в i -й узел, а c_i — строчная буква английского алфавита, задающая тип этой трубы ($1 \leq p_i \leq i - 1$).

В последующих m строках содержится информация о спецификациях, i -я из этих строк содержит разделенные пробелом целое число w_i — стоимость запуска робота в соответствии с этой спецификацией, и состоящую из строчных букв английского алфавита строку s_i — саму спецификацию ($1 \leq w_i \leq 10^9$). Суммарная длина строк s_i не превышает 10^6 .

Формат выходных данных

Первая строка выходных данных должна содержать одно число — минимальную суммарную стоимость запусков робота, в результате которых все трубы будут проверены. Если проверить все трубы невозможно, требуется вывести «-1».

Если $t = 0$, то больше ничего выводить не требуется.

Если $t = 1$ и проверить трубы возможно, то далее следует вывести список описаний запусков робота. В этом случае вторая строка выходных данных должна содержать число k — количество запусков робота, которое необходимо выполнить для проверки труб. В следующих k строках необходимо вывести по три целых числа a_i , b_i и c_i — номер узла, в котором начинается запуск, номер узла, в котором заканчивается запуск, и номер спецификации, которой соответствует запуск.

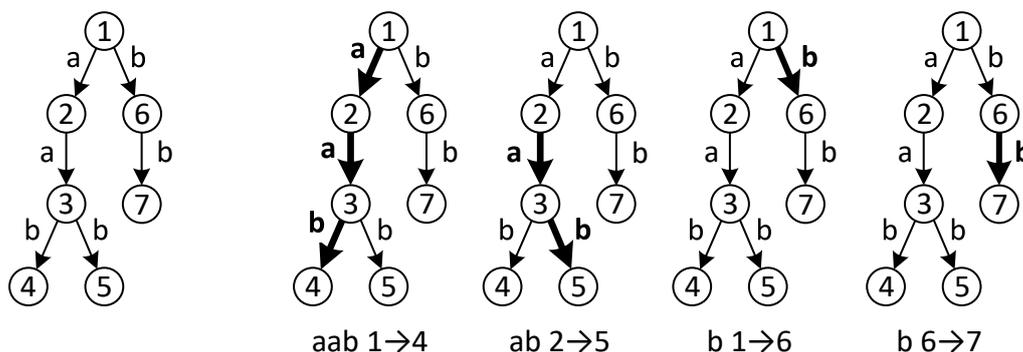
Если оптимальных способов проверки несколько, требуется вывести любой из них.

Примеры входных и выходных данных

входные данные	выходные данные
3 3 0 1 a 2 b 3 a 4 b 2 a	6
7 3 1 1 a 2 a 3 b 3 b 1 b 6 b 3 aab 5 b 2 ab	15 4 1 4 1 2 5 3 1 6 2 6 7 2

Пояснение к примеру

Система труб, заданная во втором примере входных данных, и оптимальный способ проверки всех труб для этого случая приведены на рисунке ниже.



Необходимо обратить внимание на следующие моменты:

- трубу можно проверять несколько раз, так в приведенном примере дважды проверена труба из узла 2 в узел 3;
- одну и ту же спецификацию разрешается использовать несколько раз, в приведенном примере вторая спецификация используется дважды, для проверки труб из узла 1 в узел 6 и из узла 6 в узел 7;

- робот может перемещаться по трубам только в том же направлении, по которому по трубе передается газ, спецификацию «ab» нельзя использовать для проверки труб по маршруту $2 \rightarrow 1 \rightarrow 6$, так как робот не может переместиться из узла 2 в узел 1.

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	<i>специальные условия</i>	t		
1	9	$1 \leq n \leq 500,$ $1 \leq m \leq 10^5$	Длина каждой строки s_i равна 1	$t = 0$		полная
2	10	$1 \leq n \leq 500,$ $1 \leq m \leq 10^5$	Для всех i выполнено $p_i = i - 1$	$t = 0$		полная
3	22	$1 \leq n \leq 15,$ $1 \leq m \leq 10^5$		$t = 0$		баллы
4	20	$1 \leq n \leq 500,$ $1 \leq m \leq 500$		$t = 0$		баллы
5	19	$1 \leq n \leq 500,$ $1 \leq m \leq 10^5$		$t = 0$	1 – 4	баллы
6	20	$1 \leq n \leq 500,$ $1 \leq m \leq 10^5$		$t = 1$	1 – 5	баллы

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.

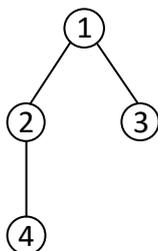


Рис. 1. Пример корневого дерева с корнем в вершине 1, листьями 3 и 4.

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.

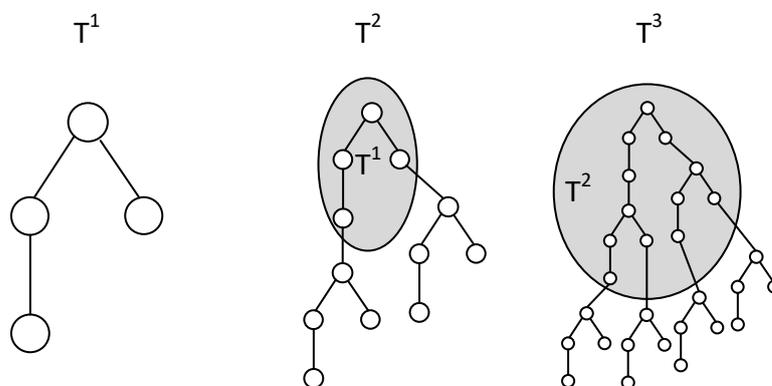


Рис. 2. Пример возведения дерева в степени 1, 2 и 3.

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.

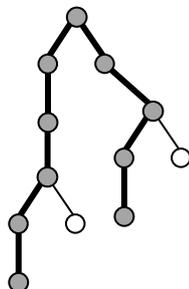


Рис. 3. Путь в дереве T^2 , содержащий максимальное количество вершин.

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