

Задача А. Алфавитно-цифровое кодирование

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В новом формате метаданных, используемых в приложении VK Video, строчные латинские буквы кодируются непрерывной последовательностью чисел от x до $x + 25$ включительно, где x — целое положительное число, являющееся кодом буквы 'a' (например, если 'a' кодируется как 97, как в стандартной кодировке ASCII, то 't' кодируется как 116, 'e' — как 101, 'c' как 99, и так далее).

Контрольный код слова определяется как сумма кодов всех букв в слове. Например, если код 'a' равен 1, то код слова 'vk' равен сумме кодов букв 'v' и 'k', то есть $22 + 11 = 33$, а код слова 'technocup' равен сумме кодов букв 't', 'e', 'c', 'h', 'n', 'o', 'c', 'u' и 'p', то есть $20 + 5 + 3 + 8 + 14 + 15 + 3 + 21 + 16 = 105$.

Вам дано значение x , то есть код буквы 'a', контрольный код слова n и индекс k . Ваша задача — вывести k -ю букву лексикографически наименьшего слова, имеющего контрольный код n , при условии, что буква 'a' имеет код x , или -1 , если соответствующего слова не существует или в нём менее k букв. Буквы в слове нумеруются слева направо, начиная с единицы.

Напомним, что слово A называется лексикографически меньшим, чем слово B , если оно либо является префиксом B , либо самая левая из несовпадающих букв в слове A стоит в алфавите раньше, чем соответствующая буква в слове B (например, 'lexmin' лексикографически меньше 'lexminimum', так как является префиксом, а 'lexmaximum' лексикографически меньше 'lexmin', так как на самой левой из несовпадающих позиций — пятой — в первом слове стоит 'a', а во втором — 'i').

Формат входных данных

Первая строка входных данных содержит одно целое число x ($1 \leq x \leq 10^9$) — код буквы 'a'. Вторая строка содержит одно целое число n ($1 \leq n \leq 10^9$) — контрольный код слова. Третья строка содержит одно целое число k ($1 \leq k \leq 10^9$) — требуемый номер позиции.

Формат выходных данных

Если слова с контрольным кодом n не существует, или же лексикографически наименьшее слово с таким кодом содержит менее k букв, выведите -1 . Иначе выведите строчную латинскую букву, которая расположена на k -м слева месте (считая, что самая левая буква расположена на первом слева месте) в лексикографически наименьшем слове, имеющем контрольный код n .

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	$x = 1$	9	—
2	$n \leq 150$	10	—
3	$n \leq 10 \cdot x$	12	—
4	$k = 1$	10	—
5	$n \leq 10^4$	29	2
6	—	30	1 — 5

Примеры

стандартный ввод	стандартный вывод
3 2 1	-1
1 2 3	-1
1 3 2	a

Задача В. Битовые операции возвращаются

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

У вас есть последовательность, состоящая из n целых неотрицательных чисел.

Пусть $\oplus$ — операция побитового «исключающего ИЛИ» (каждый бит результата равен 1, если соответствующие биты аргументов не равны между собой, и 0, если они равны, то есть бит устанавливается в 1 тогда и только тогда, когда ровно в одном аргументе соответствующий бит был равен 1). Например, $10 \oplus 6 = 1010_2 \oplus 0110_2 = 1100_2 = 12$.

Определим функцию $F(A, k)$ от последовательности целых неотрицательных чисел A и целого неотрицательного параметра k следующим образом:

- $F(A, k) = \max(A)$, если $k = 0$. То есть F равна максимальному элементу последовательности A .
- $F(A, k) = \oplus F(B, k - 1)$, если k положительно, где B — непустая подпоследовательность A . То есть F равна побитовому «исключающему ИЛИ» значений функции $F(B, k - 1)$ по всем непустым подпоследовательностям B последовательности A .

Непустая подпоследовательность — это последовательность, которая может быть получена из исходной удалением нескольких (не всех; при этом возможно, что ни одного) её элементов.

В частности, $F(A, 1)$ равна результату побитового «исключающего ИЛИ» максимумов всех непустых подпоследовательностей последовательности A .

У вас есть q запросов. В i -м запросе вам даны числа r и k . Вычислите $F(A[1..r], k)$, то есть значение функции F от префикса последовательности A длины r и параметра k .

Формат входных данных

Первая строка содержит два целых числа n и q ($1 \leq n \leq 2 \cdot 10^5$, $1 \leq q \leq 2 \cdot 10^5$), — количество элементов последовательности A и количество запросов, соответственно. В следующей строке заданы элементы последовательности — целые неотрицательные числа $0 \leq a_i \leq 10^9$. В следующих q строках заданы запросы — по два целых числа r и k ($1 \leq r \leq n$, $0 \leq k \leq 10^9$) в каждой строке.

Формат выходных данных

Для каждого запроса выведите ответ на него в отдельной строке.

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	$k = 0$ во всех запросах	10	—
2	$n \leq 20$, $q \leq 20$, $k_i = 1$	15	—
3	$n \leq 8$, $q \leq 10$, $k_i \leq 5$	20	—
4	$n, q \leq 500$	25	2
5	—	30	1 – 4

Пример

стандартный ввод	стандартный вывод
4 3	4
3 3 2 4	2
4 0	3
3 1	
1 1000000000	

Задача С. Восстановите последовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Про целочисленную последовательность известно следующее:

- Последовательность состоит из целых положительных чисел.
- Последовательность является строго возрастающей, то есть $a_i > a_j$ для всех пар $i > j$.
- Наибольший общий делитель всех элементов последовательности равен x .
- Наименьшее общее кратное всех элементов последовательности равно y .

Ваша задача — по заданным x и y найти наибольшую возможную длину такой последовательности, или определить, что таких последовательностей нет.

Формат входных данных

Входные данные состоят из нескольких тестовых примеров. Первая строка входных данных содержит одно целое число t — количество тестовых примеров ($1 \leq t \leq 10^5$).

Каждая из последующих t строк задаёт один тестовый пример и содержит два целых числа x и y ($1 \leq x \leq 10^8$, $2 \leq y \leq 10^8$).

Формат выходных данных

Для каждого тестового примера в отдельной строке выведите 0, если последовательности с вышеописанными свойствами для соответствующих x и y нет, или одно целое число — наибольшую длину такой последовательности.

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	y — простое	7	—
2	$y \leq 10^4$, $x = 1$	12	—
3	$y \leq 10^4$	13	2
4	$t \leq 1000$	20	—
5	—	48	1 – 4

Пример

стандартный ввод	стандартный вывод
2	0
2 3	2
2 4	

Задача D. Главное — любить задачи

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Все любят задачи на массивы. А ещё все любят задачи на отрезки. И, конечно же, все любят задачи на ксоры, особенно если авторы постоянно дают такие задачи. Итак, к сути.

Есть набор из n целых неотрицательных чисел $a_1, a_2, \dots, a_n$.

Введём два обозначения:

- $S(l, r) = a[l] + a[l + 1] + \dots + a[r]$
- $X(l, r) = a[l] \oplus a[l + 1] \oplus \dots \oplus a[r]$

Здесь $\oplus$ — операция побитового «исключающего ИЛИ» (каждый бит результата равен 1, если соответствующие биты аргументов не равны между собой, и 0, если они равны, то есть бит устанавливается в 1 тогда и только тогда, когда ровно в одном аргументе соответствующий бит был равен 1). Например, $10 \oplus 6 = 1010_2 \oplus 0110_2 = 1100_2 = 12$.

Посчитайте, сколько существует четвёрок целых чисел l_1, r_1, l_2, r_2 , для которых выполняются следующие три условия:

- $1 \leq l_1, l_2, r_1, r_2 \leq n$
- $l_1 \leq r_1 < l_2 \leq r_2$
- $S(l_1, r_1) + S(l_2, r_2) = X(l_1, r_1) + X(l_2, r_2)$

Так как ответ может быть очень большим, вам нужно вывести лишь остаток от его деления на $10^9 + 7$.

Формат входных данных

В первой строке дано одно целое число n ($1 \leq n \leq 10^6$).

Во второй строке даны n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i < 2^{60}$).

Формат выходных данных

Выведите одно целое число — остаток от деления количества подходящих четвёрок чисел на $10^9 + 7$.

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	$n \leq 10, a_i < 2^{30}$	7	—
2	$n \leq 100, a_i < 2^{30}$	11	1
3	$n \leq 3000, a_i < 2^{30}$	15	1 – 2
4	$n \leq 10^5, a_i < 2^{30}$	20	1 – 3
5	$a_i > 0$	19	—
6	$n \leq 3000$	12	1 – 3
7	—	16	1 – 6

Примеры

стандартный ввод	стандартный вывод
4 7 0 2 5	14
7 9 18 40 1 13 2 42	50

Задача Е. Два рациональных синуса

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Даны два рациональных числа $a = p_1/q_1$ и $b = p_2/q_2$, где $p_1, p_2 \geq 0$, $q_1, q_2 > 0$ — целые числа, такие, что наибольший общий делитель p_1 и q_1 равен 1 и наибольший общий делитель p_2 и q_2 равен 1.

Выяснить, существует ли треугольник ненулевой площади со сторонами целой положительной длины и значениями синусов двух углов, равными a и b , и, если существует, вычислить минимальное значение периметра такого целочисленного треугольника.

Формат входных данных

Первая строка входных данных задаёт число a и содержит два целых числа p_1 и q_1 ($0 \leq p_1 < 10^9$, $p_1 \leq q_1 \leq 10^9$, $q_1 > 0$, наибольший общий делитель p_1 и q_1 равен единице), такие, что $a = p_1/q_1$.

Вторая строка входных данных задаёт число b и содержит два целых числа p_2 и q_2 ($0 \leq p_2 < 10^9$, $p_2 \leq q_2 \leq 10^9$, $q_2 > 0$, наибольший общий делитель p_2 и q_2 равен единице), такие, что $b = p_2/q_2$.

Формат выходных данных

Выведите -1 , если требуемого треугольника не существует, или минимальное возможное значение периметра такого треугольника в противном случае.

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	$p_1 = q_1 = 1$ или $p_2 = q_2 = 1$	9	—
2	$p_1, p_2, q_1, q_2 \leq 10$	19	—
3	$p_1 = p_2, q_1 = q_2$	33	—
4	—	39	1 – 3

Примеры

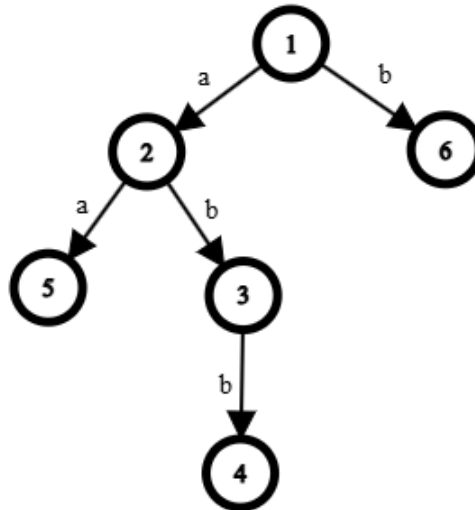
стандартный ввод	стандартный вывод
1 2 3 4	-1
1 1 3 5	12

Задача F. Есть бор? Есть два бора!

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Назовём *бором* корневое дерево, каждое ребро которого имеет *метку* в виде некоторой строчной латинской буквы. Все рёбра бора ориентированы по направлению от корня. Для любой вершины бора верно, что из неё нет двух исходящих рёбер с одинаковой меткой.

Для вершины бора v обозначим за $s(v)$ строку, состоящую из меток, выписанных с пути от корня до вершины v . Для лучшего понимания см. картинку ниже.



Корнем бора является вершина 1. $s(4) = 'abb'$, $s(6) = 'b'$, $s(3) = 'ab'$, $s(1) = ''$.

У Пети есть бор на n вершинах, пронумерованных от 1 до n . Для всех i от 1 до n Петя выписал числа a_i — вес вершины i .

У Васи тоже есть бор! Его бор имеет m вершин, пронумерованных от 1 до m . К сожалению, весов его вершины не имеют, но это не помешало ему пойти к Пете и сыграть в описанную ниже игру.

Игра состоит из q запросов. Каждый запрос состоит из k чисел $u_1, u_2, \dots, u_k$. Эти числа — номера вершин из бора Васи. Ребята составляют строку $t = s(u_1) + s(u_2) + \dots + s(u_k)$. Наконец, они считают ответ на запрос: максимально возможное значение $LCP(t, s(v_i)) \cdot a_i$ по всем вершинам v_i из бора Пети. $LCP(a, b)$ — наибольший общий префикс строк a и b . Например, $LCP('abacaba', 'ababa') = 3$, $LCP('a', 'b') = 0$, $LCP('cde', 'c') = 1$, $LCP('ab', '') = 0$.

После нескольких первых подсчётов ребята утомились играть в эту игру и пошли играть в футбол с приятелями. Вам же, как сильному программисту, необходимо доиграть эту интересную игру за них!

Формат входных данных

В первой строке дано целое число n ($2 \leq n \leq 2 \cdot 10^5$) — количество вершин в боре Пети. Корнем бора является вершина с номером 1.

В следующей строке даны n целых чисел a_i ($1 \leq a_i \leq 10^6$) — веса вершин бора Пети.

Затем в $n - 1$ строках заданы рёбра бора Пети. В i -й строке дано целое число p ($1 \leq p \leq i$) и строчная латинская буква c , обозначающие ребро из вершины p в вершину $i + 1$ с меткой c .

В следующей строке дано целое число m ($2 \leq m \leq 2 \cdot 10^5$) — количество вершин в боре Васи. Корнем бора является вершина с номером 1.

Затем в $m - 1$ строках заданы рёбра бора Васи. В i -й строке дано целое число p ($1 \leq p \leq i$) и строчная латинская буква c , обозначающие ребро из вершины p в вершину $i + 1$ с меткой c .

В следующей строке содержится целое число q ($1 \leq q \leq 2 \cdot 10^5$) — количество запросов, на которые вам необходимо ответить.

Далее в $2 \cdot q$ строках даны запросы. Каждый запрос задаётся в двух строках. В первой строке очередного запроса содержится единственное целое число k ($1 \leq k \leq 4 \cdot 10^5$). Во второй строке запроса содержатся k чисел $u_1, u_2, \dots, u_k$ ($2 \leq u_i \leq m$) — номера вершин из бора Васи.

Гарантируется, что сумма k по всем запросам не превосходит $4 \cdot 10^5$.

Формат выходных данных

Для каждого из q запросов выведите ответ на него в отдельной строке.

Система оценки

Подзадача	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	$p = i; k = 1; a_i = 1$	8	—
2	$p = i$	22	1
3	$n, m, q \leq 30$	9	—
4	$n, m, q \leq 6000$	16	3
5	$k = 1; a_i = 1$	8	1
6	$k = 1$	6	1, 5
7	$a_i = 1$	23	1, 5
8	—	8	1 – 7

Примеры

стандартный ввод	стандартный вывод
3 10 3 42 1 x 2 x 2 1 x 2 1 2 3 2 2 2	42 84
7 1 1 1 1 1 1 1 1 a 1 b 2 a 3 a 4 b 4 c 7 1 a 2 a 3 d 2 c 1 b 1 c 5 1 4 3 2 5 2 4 6 5 5 7 2 7 6 3 6 6 3	2 3 2 0 1

Задача G. Ёжик играет с графом

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

У Ёжика есть ориентированный граф на n вершинах и m ребрах. Каждая вершина обладает тремя параметрами (a_i, b_i, c_i) .

Ценой графа Ёжик называет следующую величину:

$\sum_{v=1}^n (a_v \cdot c_v + \max_u b_u)$, где максимум берется по всем вершинам u , в которые есть ребро из v . Если из v нет исходящих рёбер, считайте, что слагаемое $\max_u b_u$ равно 0.

Ёжик хочет улучшить цену графа, но просто увеличивать параметры вершин ему неинтересно. Он рассматривает k предложений по улучшению цены графа. Предложение i описывается четверкой (p_i, q_i, s_i, r_i) , здесь p_i, q_i – вершины, r_i, s_i – натуральные числа, и говорит о том, что можно выполнить одно из двух:

- Увеличить a_{p_i} на s_i
- Увеличить b_{q_i} на r_i

Чтобы Ёжику было не слишком сложно, для предложений выполнены следующие ограничения:

- Для каждой вершины v существует не более одной вершины u такой, что u присутствует среди вершин $q_1, q_2, \dots, q_k$ и из v в u есть ребро. Иными словами, не существует двух вершин $u \neq w$ среди $q_1, q_2, \dots, q_k$ таких, что из некоторой вершины v графа есть рёбра и в u , и в w .
- Каждая вершина v присутствует среди вершин $q_1, q_2, \dots, q_k$ не более G раз, где G – параметр, зависящий от подзадачи.

Крош узнал про игру Ёжика и решил усложнить ему задачу. Для этого Крош ℓ раз, пока Ёжик отвлекается, увеличивает параметр c для некоторой вершины. Именно, в j -й подход к графу Крош увеличивает c_{t_j} на d_j , где t_j – индекс изменяемой вершины, а d_j – положительное целое число ($d_j > 0$). Так как игра стала слишком сложной для Ёжика с изменениями, которые вносит Крош, вам предстоит помочь Ёжику и после каждого изменения найти максимально возможную цену графа, если сделать выбор в предложениях оптимально. Обратите внимание, что выборы в предложениях вы можете делать каждый раз заново, то есть они не зависят от выборов, сделанных ранее, а изменения остаются навсегда. Вам нужно найти максимальную цену графа также в самом начале, до подходов Кроша.

Формат входных данных

Первая строка входных данных содержит четыре целых числа n, m, k, ℓ ($2 \leq n \leq 2 \cdot 10^5$, $1 \leq m \leq 2 \cdot 10^5$, $1 \leq k \leq 2 \cdot 10^5$, $0 \leq \ell \leq 2 \cdot 10^5$), обозначающие количество вершин, количество рёбер, количество предложений и количество подходов Кроша к графу соответственно.

Вторая строка содержит n целых чисел a_i . i -е из этих чисел соответствует параметру a_i для i -й вершины ($1 \leq a_i \leq 10^3$).

Третья строка содержит n целых чисел b_i . i -е из этих чисел соответствует параметру b_i для i -й вершины ($1 \leq b_i \leq 10^9$).

Четвёртая строка содержит n целых чисел c_i . i -е из этих чисел соответствует параметру c_i для i -й вершины ($1 \leq c_i \leq 10^3$).

Каждая из последующих m строк содержит пару чисел u и v ($1 \leq u, v \leq n$, $u \neq v$) и задает ребро из вершины u в вершину v . Гарантируется, что граф не содержит кратных рёбер.

В следующих k строках содержатся предложения – k четвёрок целых чисел p_i, q_i, s_i, r_i ($1 \leq p_i \leq n$, $1 \leq q_i \leq n$, $1 \leq s_i \leq 10^3$, $1 \leq r_i \leq 10^6$). Гарантируется, что все пары (p_i, q_i) различны и что каждая вершина v присутствует среди вершин $q_1, q_2, \dots, q_k$ не более G раз, а также для

любой вершины v верно, что только одна вершина, в которую из v есть ребро, может присутствовать среди вершин $q_1, q_2, \dots, q_k$. Значение G зависит от подзадачи и определено в разделе «система оценки».

В следующих ℓ строках даны запросы – изменения Кроша. Каждый запрос состоит из двух целых чисел t_j, d_j ($1 \leq t_j \leq n, 1 \leq d_j \leq 10^4$) и обозначает, что требуется увеличить c_j на d . Все запросы перманентны, то есть изменения остаются навсегда.

Формат выходных данных

Выведите $\ell + 1$ чисел – максимально возможную цену графа до всех запросов и после каждого запроса соответственно.

Система оценки

Подзадача	$G =$	Дополнительные ограничения	Баллы	Необходимые подзадачи
1	10	$n, m \leq 2000, k \leq 10, \ell = 0$	15	—
2	5	$n, m, k \leq 2000, \ell = 0$	15	—
3	30	$\ell = 0$	27	1 – 2
4	30	—	43	1 – 4

Пример

стандартный ввод	стандартный вывод
7 6 2 2	53
3 3 2 2 3 2 2	59
2 1 3 3 6 3 3	67
1 2 1 1 2 3 2	
1 2	
1 3	
4 2	
4 5	
6 4	
7 4	
1 2 1 5	
1 4 1 3	
1 2	
1 2	

Замечание

Рассмотрим первый пример.

Изначальная цена графа равна

$$(3 \cdot 1 + \max(1, 3)) + (3 \cdot 2 + 0) + (2 \cdot 1 + 0) + (2 \cdot 1 + \max(1, 6)) + (3 \cdot 2 + 0) + (2 \cdot 3 + 3) + (2 \cdot 2 + 3) = 44$$

До запросов прибавления выгодно выбрать вторую опцию в обоих предложениях. Тогда цена станет равна 53.

После первого запроса прибавления выгодно делать такой же выбор в предложениях, а цена увеличится на 6 и станет равна 59.

После второго запроса выгодно уже выбрать первую опцию в первом предложении и вторую опцию во втором предложении. Цена станет равна 67.