Задача A. Волшебник Олег

В этой задаче достаточно разобрать два случая, вывести для каждого формулу и выбрать максимум из двух вариантов.

Если волшебную машину не покупать, то маны у Олега будет $Y \cdot T$. Во втором случае у Олега остается $Y \cdot T$ маны от первой машины, вычитается X на покупку второй, но прибавляется $Y \cdot T`$ от второй машины, где $T`$ — это время работы второй машины, оно равно T минус время, за которую Олег копит X маны на вторую машину. Значит, $T` = T - \lceil X/Y \rceil$.

Итоговый результат: $\max(Y \cdot T, Y \cdot T - X + Y \cdot (T - \lceil X/Y \rceil))$.

Критерии оценивания:

№	Баллы	Доп. ограничения	Необх. группы
1	37	$Y = 1$	–
2	31	$X = 1$	–
3	32	–	$1, 2$

Баллы выставляются автоматически проверяющей системой.

Задача B. Сколько делителей?

се общие делители некоторых чисел $a_1, a_2, \ldots, a_n$ являются делителями числа НОД$(a_1, a_2, \ldots, a_n)$. Значит, нужно посчитать число делителей НОД(x_1, x_2), НОД(x_1, x_3), НОД(x_2, x_3), но тогда трижды посчитаются делители всех трех чисел, а значит нужно дважды вычесть число делителей НОД(x_1, x_2, x_3). Число делителей числа n можно эффективно посчитать перебором до $\sqrt{n}$.

Критерии оценивания:

№	Баллы	Доп. ограничения	Необх. группы
1	11	$x_i \leqslant 10^5, x_3 = 1$	–
2	31	$x_i \leqslant 10^5$	1
3	35	$x_3 = 1$	1
4	23	–	$1 - 3$

Баллы выставляются автоматически проверяющей системой.

Задача C. Освещенность улицы

Первые группы тестов можно решить, если для каждой пары соседних фонарей a_{i-1}, a_i посчитать покрываемую ими часть улицы как $\max(k, a_i + a_{i-1})$.

Для решения последних групп тестов этого недостаточно, поскольку фонарь может покрывать не только часть улицы между соседними с ним фонарями, но и дальше. Чтобы формула выше продолжала работать, нужно передать информацию о таких фонарях соседним фонарям. Для этого зададим обновленную мощность каждого фонаря как максимум из его первоначальной мощности и наибольшей обновленной мощности его соседей слева и справа за вычетом расстояния k. После этого можно использовать решение для первых групп тестов.

Критерии оценивания:

№	Баллы	Доп. ограничения	Необх. группы
1	3	Все $a_i = 1$	–
2	16	Все $a_i \leqslant k/2$	1
3	32	Все $a_i \leqslant k$	$1, 2$
4	31	Все $a_i \leqslant 2k$	$1 - 3$
5	18	–	$1 - 4$

Баллы выставляются автоматически проверяющей системой.

Задача D. Шифр

Получим из шифра массив пар $[(k_1, s_1), (k_2, s_2), \ldots, (k_n, s_n)]$. Для этого будем добавлять в массив токенов по букве из входной строки, и будем создавать новый токен каждый раз, когда буква сменяется цифрой или наоборот.

Для позиции c_j букву можно легко определить через букву в s_i на позиции, зависимой от остатка от деления c_j на длину s_i, если известно, к какой исходной строке s_i эта буква относится. Быстро искать позицию можно через сортировку $c_1, c_2, \ldots, c_q$ — тогда можно последовательно идти по массиву пар, пока $k_i \cdot |s_i|$ не превосходит c_j, затем переходить к следующей паре.

Критерии оценивания:

№	Баллы	S	n	s_i	q	Необх. группы
1	6	$\|S\| \leqslant 10^5$	$n \leqslant 2$	$\|s_i\| = 1$	$q = 4$	–
2	13	$\|S\| \leqslant 10^5$	$n \leqslant 10^5$	$\|s_i\| = 1$	$q \leqslant 10^5$	1
3	12	$\|S\| \leqslant 10^5$	$n \leqslant 10^5$	$\|s_i\| \leqslant \|S\|$	$q \leqslant 10^5$	$1, 2$
4	15	$\|S\| \leqslant 10^{18}$	$n \leqslant 2$	$\|s_i\| = 1$	$q \leqslant 10^5$	$1, 4$
5	4	$\|S\| \leqslant 10^{18}$	$n \leqslant 2$	$\|s_i\| \leqslant \|S\|$	$q \leqslant 10^5$	$1, 4, 5$
6	13	$\|S\| \leqslant 10^{18}$	$n \leqslant 10^5$	$\|s_i\| = 1$	$q = 4$	1
7	29	$\|S\| \leqslant 10^{18}$	$n \leqslant 10^5$	$\|s_i\| = 1$	$q \leqslant 10^5$	$1, 2, 4, 6$
8	4	$\|S\| \leqslant 10^{18}$	$n \leqslant 10^5$	$\|s_i\| \leqslant \|S\|$	$q = 4$	$1, 6$
9	4	$\|S\| \leqslant 10^{18}$	$n \leqslant 10^5$	$\|s_i\| \leqslant \|S\|$	$q \leqslant 10^5$	$1 - 8$

Баллы выставляются автоматически проверяющей системой.

Задача E. Последовательное нажатие кнопок

Основная идея решения — динамическое программирование. Даже если на панели не более двух кнопок каждого типа, писать полный перебор с эмуляцией будет слишком долго, но довольно эффективно хранить для каждой кнопки с заданным числом на ней потраченную энергию на то, чтобы дойти до нее от самого старта. Посчитать ее можно, перебрав каждую кнопку с предыдущим числом, — это будет сумма энергии, которая нужна для того, чтобы со стартовой позиции дойти до перебираемой кнопки (это число нужно хранить), и расстояния между кнопками (можно посчитать как разность между коориданатами).

Проблема такого решения: его скорость, когда кнопок слишком много. В худшем случае на один пересчет будет потрачено порядка $(nm)^2$ операций. Однако для такой итерации можно заменить пересчет динамики на графовый алгоритм BFS. Его же можно использовать для решения задачи при небольших n и m.

В последней группе тестов меняется только формула для подсчета расстояния между кнопками, а также нужно использовать 0-1 BFS вместо стандартного.

Критерии оценивания:

№	Баллы	E	Доп. ограничения	Необх. группы
1	5	$E = \texttt{UDLR}$	$t = 1, n, m \leqslant 10$	–
2	19	$E = \texttt{UDLR}$	$n, m \leqslant 10$	1
3	19	$E = \texttt{UDLR}$	Все $a_{ij} \neq 0$ встречаются один раз	–
4	19	$E = \texttt{UDLR}$	Все $a_{ij} \neq 0$ встречаются не более двух раз	3
5	6	$E = \texttt{UDLR}$	Все $a_{ij} \neq 0$ встречаются не более 300 раз	$3, 4$
6	20	$E = \texttt{UDLR}$		$1 - 5$
7	12			$1 - 6$

Баллы выставляются автоматически проверяющей системой.