Задача A. Шулер

Для того, чтобы выиграть Шулеру, достаточно увеличить любое значение Васи, отличное от 6, на единицу. Единственный случай, когда это невозможно сделать — если у Васи выпали одни шестерки.

Критерии оценивания:

№	Баллы	Доп. ограничения	Необх. группы
1	13	$a = 3, b = 5, c = 4$	–
2	36	$a, b, c < 6$	1
3	51	–	$1, 2$

Баллы выставляются автоматически проверяющей системой.

Задача B. Кешбэк

Попробуем сконструировать «худший случай». Одна из таких ситуаций — это когда все покупки, кроме одной, приносят 0 бонусов, но при этом расходуют по 99 рублей. Заметим, что такой ситуации соответствует формула $\lfloor\frac{x-(n-1)\cdot 99}{100}\rfloor$. Отметим, что результат вычислений по такой формуле может быть отрицательным — в таком случае ответ будет равен 0.

Критерии оценивания:

№	Баллы	n	x	Необх. группы
1	11	$n = 1$	$x \leqslant 100$	–
2	36	$n \leqslant 2$	$x \leqslant 100$	1
3	37	$n \leqslant x$	$x \leqslant 100$	$1, 2$
4	16	$n \leqslant x$	$x \leqslant 10^{12}$	$1 - 3$

Баллы выставляются автоматически проверяющей системой.

Задача C. Быстрое решение

Ситуации в задаче независимы, поэтому для решения T ситуаций достаточно вложить в цикл `for` код решения одной ситуации.

Для групп тестов, в которых $n < 10$ достаточно научиться работать только с одной цифрой. Цифры 8 и 9 увеличить нельзя. Цифры 3 и 5 можно за одно действие увеличить до 9. Цифры 0 и 6 можно за одно действие увеличить до 8. Цифру 2 можно за два действия увеличить до 8, 4 можно за два действия увеличить до 9, а 7 за три действия можно увеличить до 9. Цифру 1 можно увеличить до 7 за одно действие, но затем следует попробовать увеличить 7 до 9, если действий достаточно.

В итоге есть набор преобразований, который можно оформить в виде последовательности `if` вида «если цифра равна некому числу и достаточно действий для увеличения этого числа, то увеличить», например, если $n = 3$, а $k \geqslant 1$, то присвоить $n = 9$.

Числа сравниваются поразрядно, а значит при $n > 10$ следует начать увеличивать число с самой левой цифры. Для этого можно циклом `for` перебрать цифры числа и максимально возможно увеличивать каждое, обновляя между шагами цикла оставшееся число операций.

Критерии оценивания:

№	Баллы	T	n	Цифры в числе n	Необх. группы
1	4	$T = 1$	$n = 9$	–	–
2	11	$T = 1$	$n < 10$	В n могут быть только $3, 9$	1
3	5	$T \leqslant 10^3$	$n = 9$	–	1
4	3	$T \leqslant 10^3$	$n < 10$	В n могут быть только $3, 5, 9$	$1 - 3$
5	17	$T \leqslant 10^3$	$n < 10$	n не содержит $1, 2, 4, 7$	$1 - 4$
6	20	$T \leqslant 10^3$	$n < 10$	В n могут быть любые цифры	$1 - 5$
7	7	$T \leqslant 10^3$	$n < 100$	В n могут быть только $3, 5, 9$	$1 - 4$
8	7	$T \leqslant 10^3$	$n < 100$	n не содержит $1, 2, 4, 7$	$1 - 5, 7$
9	9	$T \leqslant 10^3$	$n < 100$	В n могут быть любые цифры	$1 - 8$
10	4	$T \leqslant 10^3$	$n < 10^9$	В n могут быть только $3, 5, 9$	$1 - 4, 7$
11	3	$T \leqslant 10^3$	$n < 10^9$	n не содержит $1, 2, 4, 7$	$1 - 5, 7, 8, 10$
12	10	$T \leqslant 10^3$	$n < 10^9$	В n могут быть любые цифры	$1 - 11$

Баллы выставляются автоматически проверяющей системой.

Задача D. Выключить роботов

Решим задачу для $k = 1$. Построим стратегию так, что только первому роботу нужно понижать показатель отказоустойчивости полностью: выключив некого робота первым, продолжим выключение того, с кем первый имеет канал связи, затем следующего, и так далее по кругу. Пусть первый робот имеет номер t, тогда всего будет потрачено a_t секунд на этого робота и $\max(0, a_i - b_{i-k})$ для остальных роботов. Отметим, что индекс $i - k$ в формуле может оказаться нулевым или отрицательным, в таком случае имеется ввиду робот на позиции $i - k + n$. Значит, можно просуммировать все $\max(0, a_i - b_{i-k})$, а затем найти минимальную разность $a_i - \max(0, a_i - b_{i-k})$ — это робот, которого нужно выключить первым.

Задача при $k > 1$ аналогичная, но каналами связи роботы образуют НОД(k, n) отдельных «сетей» каналов. Можно явно разбить роботов по сетям каналов, а затем решить задачу способом выше для каждой сети.

Критерии оценивания:

№	Баллы	k	Доп. ограничения	Необх. группы
1	18	$k = 1$	Все $b_i = 1$	–
2	16	$k \leqslant n$	Все $b_i = 1$	1
3	38	$k = 1$	–	1
4	28	$k \leqslant n$	–	$1 - 3$

Баллы выставляются автоматически проверяющей системой.

Задача E. Порталы

Для начала решим задачу в стартовой конфигурации. При выходе со старта или из портала робот должен посетить некоторое (возможно, нулевое) количество пунктов назначения. Рассмотрим все пункты назначения между какими-то двумя порталами. Такой участок робот может пройти двумя способами: либо напрямую от одного портала к другому, либо пройти несколько левых пунктов назначения от левого портала, вернуться, и аналогично справа. Второй способ позволяет не проходить один отрезок между соседними пунктами назначения, но требует все остальные проходить дважды. Сохраним для каждого отрезка между порталами лучший из этих двух вариантов. Ответом будет сумма по всем сохраненным отрезкам плюс удвоенное расстояние от крайних порталов до крайнего пункта назначения, если они не попали между порталами.

При изменении конфигураций сохраняется некоторый подотрезок порталов, и алгоритм перемещения между порталами не меняется. Расстояние до оставшихся пунктов назначения можно посчитать, как и прежде. Поэтому две оставшиеся проблемы: скорость поиска крайних пунктов назначения и порталов для каждой конфигурации и быстрый подсчет суммы по сохраненным отрезкам между порталами. На каждую из них есть классический алгоритм: бинарный поиск и префиксные суммы соответственно.

Критерии оценивания:

№	Баллы	T	m	Необх. группы
1	12	$T = 0$	$m = 0$	−
2	43	$T = 0$	$m \leqslant 10^5$	1
3	21	$T \leqslant 10^5$	$m = 0$	1
4	24	$T \leqslant 10^5$	$m \leqslant 10^5$	$1 - 3$

Баллы выставляются автоматически проверяющей системой.