Задача A. Урок математики

Для решения задачи требуется вывести $B \cdot C + D$.

Для решения задачи на частичный балл можно заменить числа из формулы на записанные в таблице оценивания константы.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		B	C	D	
1	16	$B = 1$	$C = 7$	$D = 0$	—
2	24	$B = 1$	$C \leqslant 10$	$D = 0$	1
3	41	$B \leqslant 10$	$C \leqslant 10$	$D = 0$	1−2
4	19	$B \leqslant 10$	$C \leqslant 10$	$D < B$	1−3

Баллы выставляются автоматически проверяющей системой.

Задача B. Два варианта

Заметим, что вне зависимости от расположения точки (x, y) всё равно нужно пройти x километров, чтобы оказаться на прямой $x = 0$, где находятся дома. Затем остаётся только выбрать один из двух вариантов. Для того чтобы прийти домой, потребуется пройти ещё y километров, а чтобы прийти к другу — $|y - a|$ километров.

Во всех группах, кроме последней, количество километров до дома друга равно $x + 10 - y$. Поэтому для решения задачи на частичный балл можно упростить формулу и заменить некоторые числа константами из таблицы с ограничениями.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		x	y	a	
1	10	$x = 5$	$y = 0$	$a = 10$	—
2	26	$0 \leqslant x \leqslant 10$	$y = 0$	$a = 10$	1
3	35	$x = 5$	$0 \leqslant y \leqslant 10$	$a = 10$	1
4	16	$0 \leqslant x \leqslant 10$	$0 \leqslant y \leqslant 10$	$a = 10$	1−3
5	13	$0 \leqslant x \leqslant 10$	$0 \leqslant y \leqslant 10$	$1 \leqslant a \leqslant 10$	1−4

Баллы выставляются автоматически проверяющей системой.

Задача C. По магазинам!

Для прохождения первых четырёх групп тестов нужно пройтись при помощи цикла по строке, пока не встретятся K различных букв. Если это впервые произошло на доме номер R, то длина кратчайшего маршрута равна $2R - 1$. При реализации для первых двух групп достаточно завести две переменные, в которых сохранить количество встретившихся букв каждого типа, и проверять значения этих переменных на каждой итерации цикла. Для третьей и четвёртой группы можно использовать ту же идею, но вместо переменных завести массив длины K.

Рассмотрим пятую группу тестов. Заметим, что оптимальный маршрут полностью определяется некоторой подстрокой, содержащей Сашин дом (если длина строки равняется L, то длина маршрута равна $2L-1$). Тогда можно перебрать левую и правую границу подстроки и для каждой такой пары проверить, правда ли, что в подстроке содержатся K различных букв, при помощи двух переменных или при помощи массива длины K, что позволит пройти и седьмую группу тестов.

Для прохождения шестой группы тестов заметим, что границами подстроки могут быть только самые близкие вхождения какой-либо буквы слева и справа от дома Саши. Таким образом, сохраним в переменные индексы ближайших букв «`a`» и «`b`» слева и справа от дома Саши и посчитаем ответ разбором всех случаев — налево или направо нужно идти до магазина каждого типа.

Для полного решения задачи посчитаем расстояния до ближайших букв каждого типа справа от дома Саши и сохраним результаты в массив. Тогда, если зафиксирована длина подстроки слева от дома Саши, то необходимую длину справа можно посчитать как максимум в массиве среди всех букв, которые не встретились в подстроке слева. Решение состоит в постепенном наращивании подстроки слева с обновлением информации о том, какие буквы уже встретились слева, и вычислением того, сколько ещё нужно взять символов справа от дома Саши.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		N	K	Дополнительно	
1	8	$N \leqslant 50$	$K = 1$	s_1 = «*»	—
2	14	$N \leqslant 50$	$K \leqslant 2$	s_1 = «*»	1
3	23	$N \leqslant 50$	$K \leqslant 26$	s_1 = «*»	1 − 2
4	3	$N \leqslant 10^5$	$K \leqslant 26$	s_1 = «*»	1 − 3
5	29	$N \leqslant 50$	$K \leqslant 2$	—	1 − 2
6	7	$N \leqslant 10^5$	$K \leqslant 2$	—	1 − 2, 5
7	7	$N \leqslant 50$	$K \leqslant 26$	—	1 − 3, 5
8	9	$N \leqslant 10^5$	$K \leqslant 26$	—	1 − 7

Баллы выставляются автоматически проверяющей системой.

Задача D. Бильярд

В первых подзадачах никакие два шара не лежат на одной диагонали. Это значит, что если их столкновение возможно, то только из-за того, что один из шаров оттолкнётся от стенки и тем самым сменит диагональ, по которой он будет перемещаться. Заметим, что столкнуться тогда могут только два шара, так как для столкновения трёх и более шаров какие-то шары должны изначально находиться на одной диагонали. Таким образом, в первых четырёх подзадачах требуется установить, какие пары шаров могут столкнуться друг с другом.

Заметим, что в этом случае два шара сталкиваются, если они стартовали из одинаковой x-координаты. Тогда при помощи простого попарного сравнения x-координат можно пройти первые две группы тестов. Для прохождения третьей группы заведём массив, в котором сохраним число шаров по каждой координате x, а для прохождения четвёртой — заменим этот массив на словарь.

Если шары могут находиться на одной диагонали, то уже не обязательно пара шаров с одинаковой x-координатой столкнётся, так как на пути шара может встретиться хорошая точка.

Тогда можно разработать алгоритм, моделирующий движение шаров по моментам времени. Для прохождения пятой группы тестов алгоритм может сводиться лишь к формулам и условиям. Алгоритм, эмулирующий поведение шаров на двумерном массиве, позволяет пройти шестую группу тестов.

Заметим, что эмуляция нужна только из-за неопределённости в порядке появления хороших точек. Но заметим, что поведение шара (x, y) можно точно предсказать, если известны все хорошие точки, появляющиеся от шаров $(x`, y`)$, для которых $x` > x$ или $(x` = x$ и $y` < y)$. Действительно, только эти шары могут создать хорошие точки, в которые может попасть шар (x, y).

Таким образом, для прохождения седьмой группы тестов можно реализовать эмуляцию, в которой необходимо пускать шары в порядке уменьшения x-координаты и проверять их на столкновение с хорошими точками и другими шарами. Причём шары с одинаковой x-координатой следует пускать в порядке увеличения их y-координаты, а проверку на столкновение осуществлять только со следующим шаром в порядке такой сортировки. Проверить столкновение с хорошей точкой можно полным перебором возможных столкновений. Если для проверки эмулировать запуск шара, то такое решение пройдёт только седьмую группу тестов, а если использовать для проверки формулы, то ещё и восьмую группу.

Далее заметим, что не нужно сверять столкновение со всеми хорошими точками, если известны диагонали, по которым шар будет совершать движение. Поэтому заведём массивы, индексы которых будут являться номерами диагоналей. Положим в массивы хорошие точки. Тогда для проверки на столкновение с хорошей точкой нужно будет обратиться по соответствующему индексу в массивы. Такое решение проходит девятую группу тестов.

Для полного решения останется только заменить эти массивы на словари, чтобы работать с любыми номерами диагоналей.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		N	x_i, y_i	Дополнительно	
1	17	$N = 2$	$x_i, y_i \leqslant 50$	Никакие 2 точки не лежат на одной диагонали	—
2	4	$N \leqslant 3$	$x_i, y_i \leqslant 50$	Никакие 2 точки не лежат на одной диагонали	1
3	7	$N \leqslant 100$	$x_i, y_i \leqslant 50$	Никакие 2 точки не лежат на одной диагонали	1−2
4	9	$N \leqslant 10^5$	$x_i, y_i \leqslant 10^9$	Никакие 2 точки не лежат на одной диагонали	1−3
5	12	$N \leqslant 3$	$x_i, y_i \leqslant 50$		1−2
6	16	$N \leqslant 100$	$x_i, y_i \leqslant 50$		1−3, 5
7	11	$N \leqslant 1000$	$x_i, y_i \leqslant 500$		1−3, 5−6
8	8	$N \leqslant 1000$	$x_i, y_i \leqslant 10^9$		1−3, 5−7
9	8	$N \leqslant 10^5$	$x_i, y_i \leqslant 50\,000$		1−3, 5−7
10	8	$N \leqslant 10^5$	$x_i, y_i \leqslant 10^9$		1−9

Баллы выставляются автоматически проверяющей системой.

Задача E. Маляр

Для прохождения групп тестов с $N = 2$ отдельно рассмотрим часть промежутка, где ещё не закончились обе краски, и часть, где уже осталась только одна краска. Посчитать ответ для каждой из этих частей можно при помощи несложного разбора случаев. Идея разделять подсчёт моментами, когда какая-то краска заканчивается, пригодится для следующих групп.

Для прохождения третьей группы тестов можно полностью сэмулировать покраску забора — создать массив цветов досок в заборе. Тогда на запрос можно ответить при помощи подсчёта числа вхождений некоторого цвета на подмассиве циклом.

Для прохождения четвёртой группы тестов нужно ускорить ответ на запросы. Это можно сделать с помощью префиксных сумм — массивов, которые для каждого цвета и каждого префикса забора хранят, сколько раз этот цвет встречается на этом префиксе. Для ответа на запрос нужно вычесть из ответа для префикса r_i ответ для префикса $(l_i - 1)$. Для хранения ответов для каждого цвета потребуется довольно много памяти. Можно сократить объём необходимой памяти, считав все запросы в массив и выполняя их в порядке возрастания номера цвета. Тогда каждый раз нужно будет хранить ответы на префиксах только для того цвета, на запросы про который мы сейчас отвечаем.

В пятой группе тестов не получится создать массив, который вместит все доски. Поэтому воспользуемся идеей с множеством формул: разобьём забор на такие фрагменты, чтобы между соседними фрагментами заканчивалась какая-то краска. Такое разбиение для решения этой группы тестов можно пересчитывать для каждого запроса. На одном фрагменте количество досок нужного цвета считается как целая часть от деления суммы длины интересующего нас подфрагемента и некоторого числа, зависящего от порядкового номера интересующего нас цвета среди оставшихся красок, на количество оставшихся в наличии красок. Второе слагаемое в делимом зависит от реализации.

Для прохождения шестой группы тестов заметим, что если обрывать фрагменты не на фактическом окончании какой-либо краски, а на переходе от N-го цвета к первому, то все цвета, которые фигурируют на фрагменте, встретятся на нём одинаковое количество раз. При такой реализации можно посчитать количество цветов на каждом фрагменте и посчитать префиксные суммы по всем фрагментам. Тогда для ответа на запрос нужно обрезать интересующий нас отрезок до фрагмента, на котором интересующий нас цвет заканчивается, после чего посчитать для целых фрагментов ответ при помощи префиксных сумм, а для краёв — при помощи формулы из решения для предыдущей группы.

Для полного решения задачи осталось только ускорить формулу, так как она зависит от конкретного положения порядкового номера цвета среди всех оставшихся. Чтобы не реализовывать эту часть решения списком цветов, из которого удаляются элементы, когда заканчивается та или иная краска, можно завести структуру данных, хранящую удалённые элементы и позволяющую находить количество удалённых на отрезке. В качестве такой структуры подходят корневая декомпозиция, дерево Фенвика или дерево отрезков.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		N	Q	a_i	
1	12	$N = 2$	$Q = 1$	$a_i \leqslant 100$	—
2	9	$N = 2$	$Q \leqslant 10^5$	$a_i \leqslant 10^9$	1
3	17	$N \leqslant 1000$	$Q \leqslant 100$	$a_i \leqslant 100$	1
4	19	$N \leqslant 1000$	$Q \leqslant 10^5$	$a_i \leqslant 100$	1, 3
5	18	$N \leqslant 1000$	$Q \leqslant 100$	$a_i \leqslant 10^9$	1, 3
6	14	$N \leqslant 1000$	$Q \leqslant 10^5$	$a_i \leqslant 10^9$	1–5
7	11	$N \leqslant 10^5$	$Q \leqslant 10^5$	$a_i \leqslant 10^9$	1–6

Баллы выставляются автоматически проверяющей системой.