Задача A. Три сына

Для решения этой задачи требуется считать целые числа A, B и C, а затем вывести $A + B + C$.

Однако в этой задаче есть две вспомогательные группы тестов, позволяющие решать ее поэтапно.

Для прохождения 1-й группы тестов можно не считывать вводимые числа вовсе. Достаточно написать вывод числа 8 — ответа на задачу при $A = 8, B = 0, C = 0$.

Во 2-й группе тестов $B = 0$ и $C = 0$, поэтому достаточно считать только первое число и вывести ответ, равный A.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		A	B	C	
1	12	$A = 8$	$B = 0$	$C = 0$	—
2	25	$A \leqslant 100$	$B = 0$	$C = 0$	1
3	63	$A \leqslant 100$	$B \leqslant 100$	$C \leqslant 100$	1, 2

Баллы выставляются автоматически проверяющей системой.

Задача B. Хлопья

Для решения этой задачи достаточно написать простую проверку при помощи условного оператора, напрямую следующую из условий задачи: если $A \cdot X < B \cdot Y$, то дешевле первая коробка хлопьев, иначе — вторая.

Также эту задачу можно решить на частичный балл, если написать сравнение только чисел X и Y.

Критерии оценивания:

№	Баллы	Ограничения				Необх. группы
		A	X	B	Y	
1	56	$A = 1$	$X \leqslant 100$	$B = 1$	$Y \leqslant 100$	—
2	44	$A \leqslant 100$	$X \leqslant 100$	$B \leqslant 100$	$Y \leqslant 100$	1

Баллы выставляются автоматически проверяющей системой.

Задача C. Урок математики

В 1-й группе тестов достаточно было заметить, что если $e = 0$, то и a всегда должно быть равно 0. Тогда осталось посчитать, сколько существует b таких, что $\frac{0}{b} = \frac{0}{f}$. Понятно, что любое b в промежутке от 1 до M подходит, а значит, надо было просто вывести M.

Во 2-й группе можно было перебрать a и b, а затем проверить, что $\frac{a}{b} = \frac{e}{f}$, то есть $a \cdot f = e \cdot b$. Сложность такого решения будет $\mathcal{O}(M^2)$.

Чтобы решить 3 группу, давайте представим a в виде $k \cdot e$, где k — положительное число. Тогда $\frac{k \cdot e}{b} = \frac{e}{f}$, $k \cdot e \cdot f = e \cdot b$, а следовательно, $b = k \cdot f$.

Давайте докажем, что k в этой группе тестов должно быть целым числом. Конечно, k — рациональное (т. е. представимое в виде дроби), так как иначе число $a = e \cdot k$ было бы иррациональным, а нам нужно, чтобы a было рациональным, и более того — целым. Тогда докажем от противного: пусть k такое *нецелое* число, что $a = k \cdot e$ и $b = k \cdot f$. Так как k — рациональное, то представим k в виде несократимой дроби: $k = \frac{p}{q}$. Значит, $\frac{p \cdot e}{q}$ и $\frac{p \cdot f}{q}$ равны a и b соответственно. Но тогда e и f делятся на q, потому что p и q взаимно просты, и $q \neq 1$, так как мы взяли такое k, что оно нецелое. Следовательно, дробь $\frac{e}{f}$ можно сократить на q, но $\frac{e}{f}$ — несократимая дробь по условию группы. Противоречие. Получается, что k — целое положительное число.

Тогда если мы будем перебирать k, то a и b будут определяться однозначно. Будем увеличивать k, пока $e \cdot k$ и $f \cdot k$ меньше или равно M, и прибавлять единицу к ответу на каждом шаге перебора.

В 4-й группе может быть дана сократимая дробь. Тогда давайте просто сократим её и запустим решение для 3 группы. Для этого найдем набольший общий делитель e и f, а потом поделим e и f

на него. Наибольший общий делитель можно найти с помощью алгоритма Евклида или при помощи встроенных в стандартную библиотеку языка функций. С его оптимальной реализацией временная сложность решения будет $\mathcal{O}(M + \log \min(e, f))$. Это решение проходит первые 4 группы.

Чтобы пройти последние 2 группы, заметим, что представленный алгоритм перебора проходит ровно $M/\max(e, f)$ шагов, если $\frac{e}{f}$ — несократимая дробь. Поэтому можно сократить введенную дробь $\frac{e}{f}$ и посчитать ответ при помощи формулы. Следовательно, итоговая сложность алгоритма — $\mathcal{O}(\log \min(e, f))$ из-за алгоритма Евклида.

Критерии оценивания:

№	Баллы	Ограничения			Необх. группы
		e	M	дополнительно	
1	13	$e = 0$	$M \leqslant 1000$	—	—
2	18	$e \leqslant 10^5$	$M \leqslant 1000$	—	1
3	26	$e \leqslant 10^5$	$M \leqslant 2 \cdot 10^5$	$\frac{e}{f}$ — несократимая дробь	1
4	14	$e \leqslant 10^5$	$M \leqslant 2 \cdot 10^5$	—	1–3
5	13	$e \leqslant 10^5$	$M \leqslant 2 \cdot 10^9$	$\frac{e}{f}$ — несократимая дробь	1, 3
6	16	$e \leqslant 10^5$	$M \leqslant 2 \cdot 10^9$	—	1–5

Баллы выставляются автоматически проверяющей системой.

Задача D. Жаба Зума

Чтобы решить первые группы тестов, достаточно проверить каждый случай расстановки камней и болот. Это можно делать перебором всех вариантов входных данных (4 для первой группы и 8 для второй) при помощи условных операторов.

Полное решение задачи: будем перескакивать на ближайшую клетку того же типа, что и текущая, до которой можно добраться за 1 прыжок. В случае, если такой клетки нет, Зуме придется либо спрыгнуть в воду, либо потратить энергию, чтобы взобраться на камень — другого варианта нет.

При помощи цикла можно сымитировать этот алгоритм. На шаге алгоритма i будем считать, что Зума уже добралась до позиции i на болоте.

Однако Зума не обязана посещать все клетки. Она может некоторые клетки перепрыгивать. Чтобы это обрабатывать, заведем вспомогательную переменную `jump`. Будем считать, что `jump` $= 1$, если Зума добралась до позиции i в прыжке, то есть прыгнула с клетки $i - 1$ на клетку $i + 1$.

Начнем тело цикла с того, что проверим, установлена ли переменная `jump` в 1. Если это так, то Зума должна закончить прыжок: сделаем `jump` равной 0 и завершим итерацию цикла при помощи оператора `continue`. Так, цикл перейдет на клетку $i + 1$, и Зума окажется на следующей клетке, готовая прыгать дальше.

Иначе `jump` $= 0$, и надо прыгать. Тогда проверим $s[i + 1]$ и $s[i + 2]$ на совпадение с $s[i]$. Если $s[i + 1] = s[i]$, то запустим следующий шаг (`continue`) — на следующем шаге i увеличится на единицу, и это соответствует выбранной стратегии: Зума переместится на ближайшую клетку того же типа. Если же $s[i+2] = s[i]$, то следующий шаг надо пропустить, и сразу начать с шага $i+2$, то есть перепрыгнуть одну клетку. Для этого сделаем `jump` равной 1, и перейдем на следующий шаг (`continue`).

Будем подсчитывать вне цикла ответ в переменной e, изначально равный 0. Если Зума не может на шаге i прыгнуть на клетку того же типа, то перепрыгнем на следующую. Проверим, тратится ли на это энергия: если $s[i] = \texttt{0}$, то тратится, увеличим e на единицу. Иначе, не тратится.

Таким образом, можно написать эмуляцию, работающую за линейное время. Отметим, что в последней подгруппе потестовая система оценки, из-за чего не самые эффективные стратегии набирают некоторый частичный балл.

Критерии оценивания:

№	Баллы	Ограничения	Необх. группы
		n	
1	16	$n = 4$	—
2	24	$n \leqslant 5$	1
3	60*	$n \leqslant 10^5$	1, 2

* — в этой группе 12 тестов, каждый из которых независимо оценивается в 5 баллов.

Баллы выставляются автоматически проверяющей системой.

Задача E. Сделай равными

1 группа: Пусть $cnt(x)$ — количество чисел в массиве, равных x. Данную функцию можно посчитать при помощи линейного прохода по массиву или при помощи встроенных в стандартную библиотеку языка функций. В данной группе достаточно выбрать лучший вариант из двух: либо превратить все двойки в единицы, либо все единицы в двойки. Поэтому ответ равен $\min(n - cnt(1), n - cnt(2)) = \min(cnt(2), cnt(1))$. Сложность решения: $\mathcal{O}(n)$.

2 группа: Давайте перебирать, какому числу y из промежутка $1 \ldots k$ будут равны введенные числа в итоге. Когда число y выбрано, посчитаем необходимое число операций, чтобы сделать все числа равными y, и ответом будет минимум из таких чисел по всем y.

Если y зафиксировано, то чтобы все числа сделать равными y, надо изменить все числа a_i на y, кроме чисел, которые изначально равны y, то есть всего $n - cnt(y)$ чисел, где $cnt(y)$ можно посчитать, как в решении для прошлой группы тестов. Сложность алгоритма: $\mathcal{O}(nk)$

3 группа: Нам выгодно оставить какие-то числа без изменений. Действительно, чтобы сделать все числа равными y, надо сделать $n - cnt(y)$ операций, поэтому нам выгодно, чтобы $cnt(y)$ было как можно больше, в частности, никогда не было равно 0.

Улучшим решение предыдущей группы просто: перебирать y будем не из всех чисел, а только из $a_1, a_2, \ldots, a_n$. Считать для числа y необходимое число операций $n - cnt(y)$ будем аналогично. Сложность алгоритма: $\mathcal{O}(n^2)$

4 группа: Улучшим решение 3 группы. Долго работает линейный подсчет по массиву (то есть $cnt(y)$). Для решения этой проблемы заведем дополнительный массив h, где $h[i]$ должно быть равно $cnt(i)$. Чтобы этот массив завести за линейное время, можно запустить цикл-проход по массиву: для каждого i добавить единицу к $h[a_i]$. После этого $cnt(x)$ равно $h[x]$, и вместо подсчета $cnt(y)$ (циклом или встроенной функцией) будем обращаться к предподсчитанному ответу: $h[y]$. Сложность решения: $\mathcal{O}(n)$

Критерии оценивания:

№	Баллы	Ограничения		Необх. группы
		n	k	
1	14	$n \leqslant 1000$	$k = 2$	—
2	27	$n \leqslant 1000$	$k \leqslant 1000$	1
3	26	$n \leqslant 1000$	$k \leqslant 10^6$	1, 2
4	33	$n \leqslant 10^5$	$k \leqslant 10^6$	1–3

Баллы выставляются автоматически проверяющей системой.