Задача A (7-9). Лайки

Автор задачи:	Егор Беликов
Подготовка тестов:	Егор Беликов

Для простоты можно разобрать два случая: когда у комментария Иннокентия положительный и отрицательный рейтинги.

В случае, если итоговый рейтинг комментария положительный, то его максимально возможный рейтинг достигался бы в том случае, если бы все пользователи, кто поставил лайк, поставили бы его раньше всех тех пользователей, которые поставили дизлайк.

Вычислим, сколько было дизлайков. Всего было поставлено суммарно лайков и дизлайков N, а рейтинг комментария — R. Если бы все оценки комментария были бы положительными (все пользователи, оценившие его, поставили лайки), то рейтинг был бы равен N. Разница между настоящим и максимальным возможным рейтингом, делённая пополам ($\frac{N-R}{2}$) — это и есть количество дизлайков. Значит, максимально возможный рейтинг равен $\frac{N-R}{2} + R$.

В случае, если итоговый рейтинг комментария отрицательный, ситуация, в которой у комментария максимальный возможный рейтинг, аналогична, т.к. количество дизлайков считается по той же формуле — $\frac{N-R}{2}$, потому что $R < 0$. Значит, максимально возможный рейтинг в этом случае такой же — $\frac{N-R}{2} + R$. Эту формулу так же можно записать как $\frac{N+R}{2}$. Таким образом, данное выражение является ответом для обоих случаев.

Задача A (10-11). Лайки

Автор задачи:	Егор Беликов
Подготовка тестов:	Егор Беликов

Решение данной задачи эквивалентно решению задачи «Лайки» для 7-9 классов за исключением, что в условии задачи не гарантировалось корректность ситуации, задаваемой числами N и R.

Чтобы проверить входные данные на корректность, необходимо было сравнить чётность N и R. Для этого было достаточно проверять четность числа $N - R$: если это число нечётно, то окажется, что количество дизлайков было нецелым. Кроме того, нужно было понять, достаточное ли количество оценок дали комментарию, чтобы у него был такой рейтинг. Это можно сделать, проверив условие $|R| \leqslant N$. В случае, если хотя бы одно из условий не выполнялось, правильный ответ на задачу был равен -1.

Задача B (7-9). Обувь

Автор задачи:	Иван Корябкин
Подготовка тестов:	Марк Рябов

В данной задаче нужно было найти минимальное и максимальное количество обуви, которое мог примерить Вася. Для каждой пары обуви следовало рассмотреть несколько случаев (возможно, пересекающихся):

- если $a_i = r$, то эту модель обуви Вася точно сможет померить (к минимуму и максимуму можно прибавить 1);
- если $a_i < r$ и $b_i > 1$, то пара обуви Васиного размера может как быть на складе, так и отсутствовать в магазине (к максимуму прибавляется 1);
- если $a_i < r$ и $r - a_i + 1 = b_i$ — все размеры строго больше того, который лежит на полке в магазине присутствуют, а значит и гарантированно присутствует Васин (к минимуму прибавляется 1);
- во всех остальных случаях обуви Васиного размера гарантированно не было в наличии, а значит данная модель обуви не учитывалась при формировании ответа.

Задача C (7-9). Династии

Автор задачи:	Наиль Мингалиев
Подготовка тестов:	Наиль Мингалиев

Заметим, что если какой-то царь не является сыном другого царя, то он — правитель изначальной области. Следовательно задача свелась к поиску количества таких царей.

Заведем массив A, в котором A_i будет равняться 0, если у i-го царя указан отец, и 1 в ином случае. Тогда нам останется только посчитать сумму всех элементов этого массива, что и будет искомым ответом.

Задача D (7-9). Доска

Автор задачи: Николай Макеенков
Подготовка тестов: Николай Макеенков

Для начала заметим, что если можно вытереть какое-то множество клеток, не отрывая тряпки, то не важно, с какой из них начинать.

Тогда пусть существует грязная клетка X, вытрем ее, все грязные клетки, соседние с X, соседние с соседними с X, соседние с соседними с соседними X, и так далее. Очевидно, что при этом алгоритму не потребуется отрывать тряпку от доски, так как грязные клетки будут составлять связанную компоненту. Пройдем циклом по всем клеткам на доске: если текущая клетка не вытерта, вытрем ее и запустим из нее эту рекурсивную функцию, которая будет отмечать все соседние с ней грязные клетки очищенными.

Тогда ответом на задачу будет количество клеток, из которых мы запустили данную функцию, минус 1. Если же таких клеток не нашлось, то есть доска изначально была чистой, то необходимо вывести 0.

Задача B (10-11). День Рождения!

Автор задачи: Тимур Хисматуллин
Подготовка тестов: Тимур Хисматуллин

Рассмотрим число, записанное в k-ичной системе счисления, где a_i — цифры: $(a_s a_{s-1} a_{s-2} \dots a_1 a_0)_k = (a_s \cdot k^s + a_{s-1} \cdot k^{s-1} + a_{s-2} \cdot k^{s-2} + \dots + a_1 \cdot k + a_0)_{10}$.

Заметим следующий факт: число, заканчивающееся на 0 в k -ичной системе счисления в десятеричной делится на k. Этот факт позволяет нам понять, что любая система счисления, в которой данное число N заканчивается на 0, является делителем N.

В задаче требовалось найти минимальный делитель числа N, не являющийся 1. Существует стандартный алгоритм, который выводит все делители числа, работающий за $O(\sqrt{N})$ действий.

Стоило также обратить внимание на ограничения задачи: чтобы считать N как число, необходимо было использовать 64-битный числовой тип данных `int64` (Pascal) или `long long` (C++). Также нужно было отдельно рассмотреть случаи $N = 0$ и $N = 1$.

Задача C (10-11). Игра с конфетами

Автор задачи: Дмитрий Кузьмичев
Подготовка тестов: Дмитрий Кузьмичев

Рассмотрим сначала решение задачи на 50 баллов. Заметим, что количество конфет, которые лежат на той же горизонтали, что и рука, всегда одинаково, так как горизонталь не меняется и равна y_0. Поэтому сразу вычислим это количество. Затем будем перебирать координату x_0 положения руки (всего W вариантов). Если мы ее зафиксируем, то можем найти количество конфет, лежащих на этой вертикали, но не на горизонтали y_0: проверим для всех $1 \leqslant i \leqslant H, i \neq y_0$, есть ли в клетке (x_0, i) конфета. Всего будет рассмотрена $H - 1$ клетка, тогда итоговая сложность решения будет равна $O(WH)$.

Полное решение подразумевает аккуратный и быстрый разбор случаев. Введем обозначения: пусть нам даны прямоугольники: первый задан клетками (x_1, y_1) и (x_2, y_2), второй — клетками (x_3, y_3) и (x_4, y_4). Тогда их горизонтальные проекции на ось OX — это $[x_1; x_2]$ и $[x_3; x_4]$ соответственно, вертикальные проекции на ось OY — $[y_1; y_2]$ и $[y_3; y_4]$.

Посчитаем горизонтальную составляющую ответа. Для этого просто проверим, если y_0 лежит в пересечении вертикальных проекций, то нужно взять объединение горизонтальных проекций. Иначе, если y_0 пересекает один из прямоугольников, то берем длину его горизонтальной проекции, в противном случае горизонтальная составляющая равна 0.

Теперь вычислим вертикальную составляющую ответа. Если горизонтальные проекции пересекаются, можно взять клетку с абсциссой из этого пересечения. Еще два интересных нам случая — x_0 лежит в одной из горизонтальных проекций, то есть в $[x_1; x_2]$ или $[x_3; x_4]$. Выберем из всех этих вариантов оптимальный по количеству конфет, а при равенстве — тот, у которого абсцисса минимальна. Нужно не забывать контролировать то, что точка (x_0, y_0) посчитана не более одного раза. Легко видеть, что такое решение работает за константое время, то есть имеет сложность $O(1)$.

Задача D (10-11). Конкатенация

Автор задачи: Егор Луничкин
Подготовка тестов: Егор Луничкин

- **Решение за** $O(N^3)$ — 30 баллов.

  Чтобы набрать 30 баллов, достаточно было перебрать всевозможные тройки чисел и для каждой из них проверить, выполняется ли условие задачи. Несколько больше баллов можно было набрать, предварительно отсортировав массив любой известной вам сортировкой, так как длины хотя бы двух чисел из подходящей тройки будут не больше половины максимальной длины числа. Велика вероятность того, что эти числа будут находиться в начале отсортированного массива.

- **Решение за** $O(N^2 \log N)$ — 70 баллов.

  Заметим, что перебирать тройки чисел излишне. Когда мы зафиксировали какие-то два числа, третье задаётся почти однозначно: мы можем приписать первое число ко второму справа или слева. Нам осталось научиться быстро узнавать, есть ли хоть одно из этих чисел в массиве. Для этого можно воспользоваться бинарным поиском по отсортированному массиву (массив нужно предварительно отсортировать). В данной подзадаче это можно сделать любой известной вам сортировкой, в том числе квадратичной.

- **Решение за** $O(N \log N)$ — 100 баллов.

  Теперь заметим, что каждое из чисел в нашем массиве не превосходит 10^{18}, а значит не длиннее 19 символов. Значит, разделить какое-то число на два других мы можем не более чем 18 способами. Будем идти по массиву и для каждого числа перебирать всевозможные разрезы. Зафиксировав какой-то разрез, мы получаем два числа и проверям, есть ли они в массиве (например, снова бинарным поиском). Однако теперь предварительная сортировка массива должна быть быстрее, чем квадратичная. Можно воспользоваться быстрой сортировкой, сортировкой кучей, сортировкой слиянием или любой другой, работающей за $O(N \log N)$.