Задача A. Астронавигация

В случае чётного n ответ равен n: например, соединим планеты $2k$ и $2k-1$ для всех k от 1 до $n/2$. В случае нечётного n все n пожеланий выполнить не получится: будем добавлять рейсы по одному и считать суммарное количество рейсов в космопортах всех планет. Каждый рейс увеличивает это количество на 2, изначально количество рейсов было 0, то есть сумма всегда будет чётной. А в случае, если все n пожеланий выполнены, сумма будет нечётной. $n-1$ пожелание при нечётном n можно выполнить, например, соединив планету 1 рейсами с каждой из оставшихся планет.

Таким образом, решение — прочитать число как строку, и, если последний байт является нечётным, уменьшить его на 1.

Задача B. Беспосадочные перелёты

Посчитаем для каждого из рейсов разность между временем прилёта и временем вылета. В одном случае она равна $t1-tz$, а в другом $t2+tz$, где $t1$ и $t2$ — чистое время в пути для каждого из рейсов, а tz — разность времени в часах между городами. Тогда при вычитании из большей разности меньшей получаем $2tz+(t1-t2)$. Заметим, что по условию $t1-t2$ строго меньше одного часа, а tz — целое число часов, так что значение tz можно восстановить однозначно (разность времени в минутах, делённая на 60 нацело, делённая на 2 и округлённая вверх), после чего восстанавливается и время в пути.

Задача C. Выбери звезду

Из условия задачи прямо следует, что звено ломаной — это некоторая диагональ правильного n-угольника. Задача сводится к тому, чтобы найти максимальную диагональ со следующим свойством: переходя по диагоналям соответствующей длины, мы вернёмся в исходную точку только после n переходов.

Занумеруем вершины целыми числами от 0 до $n-1$, тогда переход по каждой

Задача D. Гигантский аэробус

Рассмотрим общее количество «выгодных» рассадок. Очевидно, что все ряды и все группы кресел являются независимыми. Обозначим занятое место нулём, свободное место — единицей, получаем, что количество выгодных рассадок в группе размера k_i равно количеству последовательностей из k_i нулей и единиц, в которой нет двух единиц подряд.

С помощью динамического программирования несложно доказать, что таких последовательностей будет F_{k_i}, где F_x — x-е число Фибоначчи. Таким образом, общее количество вариантов равно произведению F_{k_i} в n-й степени.

Рассмотрим количество рассадок, являющихся «выгодными», но не «приемлемыми». В аналогичной модели получается, что для одного ряда это количество последовательностей из $s=\sum\limits_1^g k_i$ нулей и единиц, в которой нет двух единиц подряд, получаем соответствующее число Фибоначчи F_s. Общее число таких рассадок равно F_s в n-й степени.

Таким образом, ответ равен разности первого и второго количеств. Заметим, что для ускорения решения задачи числа Фибоначчи от 1 до $100\cdot 100=10^4$ лучше предпросчитать по соответствующему модулю.

Задача E. Древняя система счисления

Сгенерируем все римские числа, при этом для каждого подсчитаем произведение его цифр в виде степени пятёрки (она будет не более, чем третьей) и количества нулей после неё, хранить будем как строку. После чего заводим `map <string, pair <int, int»`, в котором для каждой строки-произведения поддерживаем наибольшее и наименьшее числа с таким произведением.

Далее на каждый запрос просто выводим соответствующее значение по ключу в виде числа из запроса, прочитанного как строка, или -1, если ключа в `map` нет.