Задача A. Стакан

В этой задаче требовалось просто сложить три целых числа.

Задача B. Четыре чертенка

Проверить, что два каких-то введенных числа равны, можно при помощи довольно громоздкого `if`. Но можно подумать, как упростить условие для проверки, если единица, двойка и тройка гарантированно встретятся среди введенных чисел.

Зафиксируем мысленно трех чертят, которые заняты разными частями чертежа. Посмотрим на действия последнего (четвертого) чертенка. Если он ничего не делает, то ответ — «`NO`». Если же он чертит любую из трех частей, то он это делает совместно с одним их трех зафиксированных чертят, тогда ответ — «`YES`».

Таким образом, ответ «`NO`» достигается только в том случае, если среди введенных чисел не встречается ноль. А так как все числа по условию неотрицательные, условие можно упростить до: `if min(a1, a2, a3, a4) == 0`.

Задача C. Пицца

Пусть Боря приведет N друзей. Чтобы всем досталось поровну, должно существовать такое d — количество кусков для каждого из пришедших, что $N \cdot d = K$. Тогда K должно делиться на N. Значит, нужно при помощи цикла найти максимальное нечетное число N, которое делит K.

Бонус: Как решать эту задачу при $K \leqslant 10^{18}$?

Задача D. Мем ли?

Задача состоит из двух частей.

Первая: посчитать количество заглавных и строчных букв. Это можно сделать при помощи прохода циклом `for` по строке с `if` внутри цикла. Отметим, что для проверки регистра необязательно было использовать встроенные функции или сравнивать коды символов. Вместо этого можно было написать длинный `if`, перебирающий все возможные варианты заглавных букв, которых по условию всего четыре: «L», «O», «K» и «E».

Вторая: проверить условие. Для этого следует воспользоваться функцией взятия модуля числа. Пусть a — количество заглавных букв, а b — строчных. Тогда достаточно проверить, что $|a - b| \leqslant 1$.

Задача E. Дураки и дороги

Подсказка: Как отправить человека от некоторой точки максимально далеко?

Интуиция подсказывает, что если на каждом участке пути отправлять Леху максимально далеко, то достигается худший случай. Таким образом, кандидат на худший случай — это путь $n, 1, (n-1), 2, \ldots$. Оказывается, это действительно так.

Длину такого пути легче посчитать, если разбить все отрезки на несколько групп и просуммировать длины в каждой из них по отдельности. Простой пример: сразу добавить первый шаг $a_n - a_1$, а потом посчитать шаги влево: $(a_n - a_1) + (a_{n-1} - a_2) + \ldots + (a_{n-i} - a_{i+1}) + \ldots$, а затем вправо: $(a_{n-1} - a_1) + (a_{n-2} - a_2) + \ldots + (a_{n-1-i} - a_{i+1}) + \ldots$.

Также из-за того, что $n \leqslant 8$, пути можно было перебирать при помощи перебора перестановок.

Почему придуманный путь действительно самый длинный? Для этого посмотрим на отрезки между соседними домами. Пусть слева от него a домов, а справа – b. Заметим, что мы по каждому такому пройдемся не больше раз, чем $\min(2 \cdot a + 1, 2 \cdot b)$ — это следует из того, что через этот отрезок можно перемещаться только в дом слева от него, в дом справа или наоборот. С другой стороны, данный путь всегда проходится по таким отрезкам ровно столько раз, сколько возможно. Таким образом, это самый длинный путь.

Бонус: Это не единственный путь, на котором достигается ответ. Найдите такой путь, что его длина — максимально возможная, но максимальное перемещение между двумя домами на этом пути — минимально возможное среди всех таких путей.

Задача F. Упрощенный боулинг

Подсказка 1: Можно ли решить эту задачу вообще без десяток?

Подсказка 2: В задаче два случая с $k > 90$ и $k \geqslant 90$.

Постараемся избавиться от особенности начисления очков при десяти кеглях.

Можно никогда не ставить десятки. Для этого можно явно ограничить каждое число девятью, и выводить минимум из 9 и оставшихся нераспределенных очков, которые нужно добавить в протокол. Таким методом можно получить до $9 \cdot 10 = 90$ очков, что для решения задачи недостаточно.

С другой стороны, когда $k > 90$ можно поставить несколько десяток, а в оставшиеся клетки расставить девятки. Например, начнем с четырех десяток и девятки. Тогда сумма за первые пять раундов будет 88. Осталось за пять раундов добрать еще очков, используя числа, не больше девяти. Можно получить таким образом все результаты до 133, что нам достаточно.

Бонус: Как решить задачу при $k \leqslant 190$? С такими ограничениями не для всех k гарантируется, что существует вообще какой-либо протокол, по которому результат — k баллов.

Задача G. Подпоследовательности

Посмотрим, как будет меняться ответ для функций с шагом аргумента в 2. Например, возьмем $f(2)$, $f(4)$, $f(6)$, ...

Шаг в два в этой задаче полезен тем, что итоговое количество нечетных чисел в выбранной подпоследовательности должно быть четно. Так, пусть мы знаем ответ для $f(q)$, и знаем, какие числа были использованы в подпоследовательности для получения максимальной суммы. Тогда $f(q+2)$ — это число $f(q)$, к которому нужно прибавить либо два наибольших четных числа, либо два наибольших нечетных числа.

Действительно, убирать какое-то количество чисел из уже взятых бессмысленно, так как из-за того, что для получения ответа $f(q)$ была взята подпоследовательность длины q с максимально возможной суммой, замена гарантированно приведет к уменьшению суммы и, как следствие, к не самой лучшей подпоследовательности.

Случай с нечетными значениями отличается только тем, что $f(1)$ требуется сделать равным максимальному четному числу.

Таким образом, в решении нужно оптимально искать максимум и минимум среди четных и нечетных, что можно сделать при помощи вспомогательных массивов и сортировки.