Задача A. Стакан

В этой задаче требовалось просто сложить три целых числа.

Задача B. Боря

Одну булку всегда берет Боря, а оставшиеся $K-1$ людей разбиваются на $\frac{K-1}{2}$ пар, каждой из которой тоже нужно взять по булке. Таким образом, ответ: $1+\frac{K-1}{2}$.

Задача C. 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 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 может быть две: «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. Убург

Подсказка 1: перенумеруйте дома от 1 до 12.

Подсказка 2: даже на маленьких числах имеет смысл написать `for`.

Заметим, что вам даны часы, циферблат которых разбит на 3 сектора. Действительно, можно перейти от задачи с городом к задаче про циферблат, если определить час на старте как $c = 4 \cdot (a-1)+x$ и час на финише как $d = 4 \cdot (b-1) + y$. Теперь требуется посчитать, сколько цифр пройдет стрелка от часа c к часу d.

Такую задачу можно решить двумя способами. Первый — смоделировать движение стрелки при помощи цикла. Для этого просто будем прибавлять на каждой итерации цикла единицу к c, если $c < 12$, а иначе менять значение c с 12 на 1. Таким образом, c будет показывать в цикле час, на который в текущий момент показывает стрелка часов. Для ответа нужно также держать счетчик, через сколько цифр уже прошла стрелка, и на каждой итерации прибавлять к нему единицу.

Другой подход — заметить, что если $d > c$, то ответ на задачу: $d-c+1$. Тогда если c и d поменять местами, то будет посещена оставшаяся часть циферблата, а также стартовые и конечные точки. Поэтому если $d < c$, то ответ равен $n-(d-c+1)+2 = n-d+c+1$.

Задача F. Упрощенный боулинг

Подсказка: Если с десятками сложно, как решить эту задачу вообще без десяток?

Постараемся избавиться от особенности начисления очков при десяти кеглях.

Первая возможная идея: 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 достаточно.

Другой вариант с той же идеей: распределить очки между всеми раундами примерно поровну. Можно показать, что при $k \leqslant 50$ выводимые числа не превысят 6.

Вторая возможная идея: можно писать протокол через одну клетку. Тогда даже если в протоколе встретится десятка, на итоговое число очков это не повлияет — оно тоже будет считаться как сумма всех чисел в протоколе. Для решения задачи будем для каждой пары соседних раундов выводить первым числом минимум из 10 и оставшихся нераспределенных очков, а вторым — ноль. Это решение как раз может сгенерировать протоколы при $k \leqslant 50$.

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. 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.