

А. Простой номер

Ограничения позволяют перебрать все возможные разбиения строки на подстроки. Если каждая подстрока в каком-либо разбиении является простым числом, то ответ "YES", а если такого разбиения найти не удалось, то ответ "NO".

Для определения простоты числа можно воспользоваться "Решетом Эратосфена" - сначала найти все простые числа меньше 1 000 000, а затем для определения простоты конкретного числа перебирать простые делители, меньшие корня квадратного этого числа.

Отдельно стоит отметить, что числа в разбиении могут быть с ведущими нулями, так как в условии явно не указано обратное (то есть 002 == 2).

В. Самый популярный язык программирования

Сначала нужно сгруппировать языки программирования по количеству упоминаний во входных данных. Затем отсортировать по убыванию количества упоминаний. Если языков больше одного и в отсортированном списке у первого и второго одинаковое количество упоминаний, то ответ "No winner", иначе нужно вывести название первого языка.

С. Коробка в коробке

Одним из вариантов решения задачи может быть построение ориентированного графа, каждое ребро которого $E(A, B)$ означает, что коробка с номером A помещается в коробку с номером B . Получившийся граф является набором деревьев (то есть не имеет циклов), так как не существует двух коробок A и B таких, что A помещается в B и B помещается в A . Ответом в этом случае будет самый длинный путь в графе.

Для нахождения пути можно из каждой вершины запустить поиск в глубину, запоминая на выходе из вершины максимальную длину пути, который начинается в этой вершине. Для восстановления пути нужно обойти вершины, начиная с вершины с максимальной длиной пути, переходя каждый раз к вершине с длиной пути, меньшей на 1.

D. Кубики

Общее количество секунд, которое потребуется на окрашивание всех кубиков:

$$C = 6NM + 2NK$$

Количество секунд в ответе:

$$S = C \bmod 60$$

Количество минут:

$$M = (C \operatorname{div} 60) \bmod 60$$

Количество часов:

$$H = C \operatorname{div} 3600$$

div - операция целочисленного деления

mod - операция взятия остатка от целочисленного деления

E. Степень палиндрома

Рассмотрим последовательность проверок подстрок данной строки.

Если вся строка является палиндромом, то степень равна 1.

Если половина ($\frac{1}{2}$ длины) строки является палиндромом, то степень равна 2.

Если четверть ($\frac{1}{4}$ длины) строки является палиндромом, то степень равна 3.

И так далее, пока длина проверяемой строки не станет равна 1 (строка из одного символа всегда является палиндромом).

Оценка сложности этого алгоритма $O(n \log n)$.

F. Растеризация

Сначала нужно построить изображения каждого символа из шрифта, то есть растеризовать шрифт (строить изображения можно и перед непосредственным размещением символа). Зная как будет выглядеть символ шрифта (а именно его самую верхнюю и самую нижнюю точки), можно вычислить его положение по вертикали. Положение по горизонтали можно вычислить, если начиная от крайней левой координаты предыдущего символа двигать символ вправо на одну позицию до тех пор, пока не выполнятся все условия размещения (т.е. он не будет касаться уже размещенных символов). Для вывода нужно вычислить прямоугольную область, которую занимает нарисованный текст (то есть крайние левую, правую, верхнюю и нижнюю точки).

G. Непростой путь

Можно обойти карту, начиная с начальной точки начала маршрута, поиском в ширину. При первом посещении свободной клетки запоминать длину маршрута, который в нее привел.

При выводе карты для каждой свободной клетки проверить, является ли длина маршрута, который в нее привел, простым числом. Простые числа можно найти при помощи “Решета Эратосфена”.

Также нужно помнить, что на карте могут быть недостижимые клетки, с длиной маршрута 0.

H. Равные треугольники

Для проверки достаточно отсортировать списки длин сторон по возрастанию (или убыванию). Если отсортированные списки равны, то треугольники равны.

I. Поворот

Сначала нужно установить “указатель” на положение буквы **a** в строке, нанесенной на диск. Затем для каждого символа в шифруемой строке (кроме пробела - его нужно пропускать) нужно найти символ на диске, смещая “указатель” на один символ вправо (то есть добавляя 1), и перенося его на 0 при выходе за пределы строки. Количество смещений будет значением шифра для данного символа.

Также количество смещений можно найти при помощи формулы, если предварительно сохранить положение всех букв нанесенных на диск.

$$X_i = (26 + Pos_i - Pos_{i-1}) \bmod 26$$

X_i - значение шифра для текущего символа

Pos_{i-1} - положение на диске предыдущего символа (для первого символа 0)

Pos_i - положение на диске текущего символа