**ЗАДАНИЯ**

**школьного этапа Всероссийской олимпиады**

**школьников по информатике в 2019/2020 уч. году.**

**9 - 11 классы**

**Максимальное количество баллов – 100.**

**Время выполнения – 2 часа (120 минут).**

**Задача 1.** Забавная игра (20 баллов)

|  |  |
| --- | --- |
| Имя входного файла: | d.in |
| Имя выходного файла: | d.out |
| Максимальное время работы на одном тесте: | 3 секунды |
| Максимальный объем используемой памяти: | 8 мегабайт |

Легендарный учитель математики Юрий Петрович придумал забавную игру с числами. А именно, взяв произвольное целое число, он переводит его в двоичную систему счисления, получая некоторую последовательность из нулей и единиц, начинающуюся с единицы. (Например, десятичное число 1910 = 1⋅24+0⋅23+0⋅22+1⋅21+1⋅20 в двоичной системе запишется как 100112.) Затем учитель начинает сдвигать цифры полученного двоичного числа по циклу (так, что последняя цифра становится первой, а все остальные сдвигаются на одну позицию вправо), выписывая образующиеся при этом последовательности из нулей и единиц в столбик — он подметил, что независимо от выбора исходного числа получающиеся последовательности начинают с некоторого момента повторяться. И, наконец, Юрий Петрович отыскивает максимальное из выписанных чисел и переводит его обратно в десятичную систему счисления, считая это число результатом проделанных манипуляций. Так, для числа 19 список последовательностей будет таким:

10011

11001

11100

01110

00111

10011

и результатом игры, следовательно, окажется число 1⋅24+1⋅23+1⋅22+0⋅21+0⋅20 = 28.

Поскольку придуманная игра с числами все больше занимает воображение учителя, отвлекая тем самым его от работы с ну очень одаренными школьниками, Вас просят написать программу, которая бы помогла Юрию Петровичу получать результат игры без утомительных ручных вычислений.

Формат входных данных

Входной файл содержит одно целое число *N* (0≤*N*≤32767).

Формат выходных данных

Ваша программа должна вывести в выходной файл одно целое число, равное результату игры.

Пример

|  |  |
| --- | --- |
| d.in | d.out |
| 19 | 28 |

Задача 2. Часы с боем (20 баллов)

|  |  |
| --- | --- |
| Имя входного файла: | a.in |
| Имя выходного файла: | a.out |
| Максимальное время работы на одном тесте: | 3 секунды |
| Максимальный объем используемой памяти: | 64 мегабайта |
|  |  |

Старинные часы бьют каждые полчаса. Причем в начале каждого часа они бьют столько раз, сколько сейчас часов (по 1 разу – в час ночи и в час дня, по 2 раза – в два часа ночи в два часа дня и т.д., в полночь и в полдень они бьют, соответственно, по 12 раз). И еще 1 раз они бьют в середине каждого часа.

Дан промежуток времени (известно, что прошло строго меньше 24 часов). Напишите программу, определяющую, сколько ударов сделали часы за это время.

Формат входных данных

В первой строке записан начальный момент времени, во второй строке — конечный. Моменты времени задаются двумя целыми числами, разделяющимися пробелом. Первое число задает часы (от 0 до 23), второе — минуты (от 1 до 59, при этом оно не равно 30).

Формат выходных данных

В выходной файл выведите одно число — сколько ударов сделали часы за этот отрезок времени.

Примеры

|  |  |
| --- | --- |
| a.in | a.out |
| 5 20  10 25 | 45 |
| 10 25  5 20 | 135 |
| 5 2   1. 21 | 0 |

Задача 3. Выборы жрецов (30 баллов)

|  |  |
| --- | --- |
| Имя входного файла: | b.in |
| Имя выходного файла: | b.out |
| Максимальное время работы на одном тесте: | 3 секунды |
| Максимальный объем используемой памяти: | 64 мегабайта |
|  |  |

В стране Олимпиадии снова выборы.

Страна состоит из маленьких графств. Графства объединяются в конфедерации. Каждая конфедерация раз в год выбирает себе покровителя – одного из 200 жрецов. Этот ритуал называется Великими Перевыборами Жрецов и выглядит так: конфедерации одновременно подают заявления (одно от конфедерации) в Совет Жрецов о том, кого они хотели бы видеть своим покровителем (если заявление не подано, то считают, что конфедерация хочет оставить себе того же покровителя). После этого все заявки удовлетворяются. Если несколько конфедераций выбирают одного и того же Жреца, то они навсегда объединяются в одну. Таким образом, каждый Жрец всегда является покровителем не более чем одной конфедерации. Требуется написать программу, позволяющую Совету Жрецов выяснить номер Жреца-покровителя каждого графства после Великих Перевыборов. В Совете все графства занумерованы (начиная с 1). Все Жрецы занумерованы числами от 1 до 200 (некоторые из них сейчас могут не быть ничьими покровителями).

Формат входных данных

Во входном файле записано число *N* – количество графств в стране (1≤*N*≤5000) – и далее для каждого графства записан номер Жреца-покровителя конфедерации, в которую оно входит (графства считаются по порядку их номеров). Затем указаны заявления от конфедераций. Сначала записано число *M* – количество поданных заявлений, а затем *M* пар чисел: первое число – номер текущего Жреца-покровителя, второе – номер желаемого Жреца-покровителя.

Все числа во входном файле разделяются пробелами и (или) символами перевода строки.

Формат выходных данных

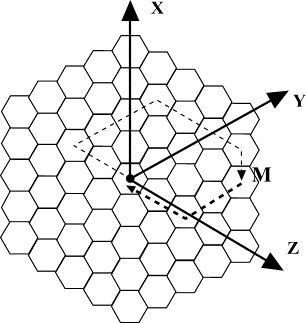
В выходной файл вывести для каждого графства одно число – номер его Жреца-покровителя после Великих Перевыборов. Сначала – для первого графства, затем – для второго и т.д.

Пример

|  |  |
| --- | --- |
| b.in | b.out |
| 7  1 1 5 3 1 5 1  2  5 1  1 3 | 1. 3 1 3 3 1 3 |

Задача 4. Гексагон (30 баллов)

|  |  |
| --- | --- |
| Имя входного файла: | d.in |
| Имя выходного файла: | d.out |
| Максимальное время работы на одном тесте: | 3 секунды |
| Максимальный объем используемой памяти: | 64 мегабайта |
|  |  |

Поле для игры в новую игру "Гексагон" разбито на шестиугольники (см. рисунок). Игрок, стартуя из некоторого начального шестиугольника, сделал несколько ходов. Каждый ход заключается в перемещении фишки в соседний шестиугольник (имеющий с тем, где находилась фишка до начала хода, общую сторону) — тем самым, ход делается вдоль одного из направлений X, Y или Z (см. рисунок). Игрок записал все свои ходы, причем если фишка двигалась вдоль какого-либо направления несколько раз подряд, то в записи это обозначается указанием направления и количества ходов, которые были сделаны.

Напишите программу, которая найдет кратчайший (по количеству совершаемых ходов) путь в начальную клетку из той, где фишка оказалась после ходов игрока.

Формат входных данных

В первой строке входного файла записано число *N* — количество строк в записи перемещений фишки (1≤*N*≤100). Далее идет *N* строк с записью ходов: в каждой строке записана сначала большая буква X, Y или Z, задающая направление, затем пробел, и число, задающее количество ходов в данном направлении (число может быть и отрицательным, если игрок перемещал фишку параллельно оси, но в направлении, противоположном направлению оси). Все числа по модулю не превышают 200.

Формат выходных данных

В выходной файл выведите описание кратчайшего пути обратно в начальную клетку в том же формате, в каком описание задано во входном файле (за исключением ограничений). Все числа, определяющие количество ходов в каком-либо направлении, должны быть ненулевыми.

Пример

|  |  |
| --- | --- |
| d.in | d.out |
| 4  Z -2  Y 3  Z 3  X -1 | 2  Y -2  Z –2 |