

1. Настенные часы

1 секунда

30 очков

В кабинете мэра города висят настенные часы, у которых часовая и минутная стрелка двигаются с постоянной скоростью. Поскольку минутная стрелка длиннее, она иногда полностью закрывает часовую, и мэр, глядя на часы, видит только одну стрелку.

Написать программу, которая подсчитывает, сколько раз это событие происходит в заданном интервале времени.

Входные данные. В первой строке текстового файла `sk.sis` задано начало временного интервала в виде $TT:MM$ ($00 \leq TT \leq 23$, $00 \leq MM \leq 59$), и во второй строке — конец интервала в том же формате.

Выходные данные. В единственную строку текстового файла `sk.val` вывести целое число K , показывающее, сколько раз стрелки встречаются в заданном интервале времени. Начало и конец интервала считаются частью интервала (то есть если стрелки встретятся ровно в $TT:MM$, это событие тоже нужно посчитать).

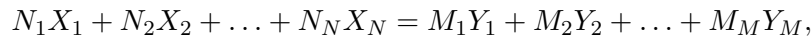
Пример.	<code>sk.sis</code>	<code>sk.val</code>
	00:00	12
	12:00	

2. Уравнение реакции

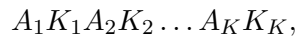
1 секунда

30 очков

Для обозначения химических реакций используют уравнения реакций вида



где показано, что N_1 молекул вещества X_1 , N_2 молекул вещества X_2 , ..., N_N молекул вещества X_N реагируют между собой, и в результате получается M_1 молекул вещества Y_1 , M_2 молекул вещества Y_2 , ..., M_M молекул вещества Y_M . Коэффициенты $N_1, N_2, \dots, N_N, M_1, M_2, \dots, M_M$ — целые числа, а вещества $X_1, X_2, \dots, X_N, Y_1, Y_2, \dots, Y_M$ в свою очередь представляются формулами вида



что значит, что одна молекула этого вещества состоит из K_1 атомов элемента A_1 , K_2 атомов A_2 , ..., K_K атомов A_K . В этих формулах коэффициенты K_1, K_2, \dots, K_K — также целые числа. Химические элементы обозначаются сокращениями из латинских букв. Коэффициенты, равные 1, опускают как в уравнении реакции, так и в формуле вещества.

Например, уравнение реакции $C_2H_5OH + 3O_2 = 2CO_2 + 3H_2O$ показывает, что одна молекула этанола (в которой два атома углерода, пять атомов водорода, один атом кислорода и ещё один атом водорода) реагирует с тремя молекулами кислорода (в каждой из которых два атома кислорода), и в результате этой реакции возникает две молекулы углекислого газа (в каждой из которых один атом углерода и два атома кислорода) и три молекулы воды (в каждой — два атома водорода и один атом кислорода).

Говорят, что уравнение реакции уравновешено, если в его правой и левой частях в сумме находится одинаковое количество атомов каждого элемента. Например, приведённое уравнение реакции горения этанола уравновешено, так как в его левой части в одной молекуле этанола и трёх молекулах кислорода в сумме 2 атома углерода, $5 + 1 = 6$ атомов водорода и $1 + 3 \cdot 2 = 7$ атомов кислорода, а в правой части в двух молекулах углекислого газа и трёх молекулах воды в сумме также 2 атома углерода, $2 \cdot 2 + 3 = 7$ атомов кислорода и $3 \cdot 2 = 6$ атомов водорода.

В то же время уравнение $H_2 + O_2 = H_2O$ неуравновешено: хотя атомов водорода в обеих частях одинаковое количество, атомов кислорода в левой части два, а в правой части только один.

Написать программу, которая проверяет, уравновешено ли данное уравнение химической реакции.

Входные данные. В первой строке текстового файла `rv.sis` дано количество изучаемых уравнений N ($1 \leq N \leq 10$), и в каждой из следующих N строк задано одно уравнение. Можно предполагать, что длина каждого уравнения не превышает 100 символов и все уравнения синтаксически корректны. Можно также предполагать, что все коэффициенты состоят из одной цифры и обозначения всех химических элементов состоят из одной большой латинской буквы.

Выходные данные. В текстовый файл `rv.val` вывести ровно N строк: в i -ю строку вывести слово `JAH`, если уравнение в $(i + 1)$ -ой строке входного файла уравновешено, и слово `EI`, если нет.

Пример.	<code>rv.sis</code>	<code>rv.val</code>
	2	JAH
	<code>C+O2=C02</code>	EI
	<code>H2+O2=H2O</code>	

3. Возведение матриц в степень

1 секунда

40 очков

Для математического анализа связей между некоторыми объектами их часто представляют в виде квадратных таблиц, где строки и столбцы представляют изучаемые объекты, а значение на пересечении i -ой строки и j -го столбца показывает, находятся ли объекты i и j в этой связи (истина или ложь).

При исследовании таких связей иногда оказывается полезным умножение таких таблиц. Произведение двух таблиц с N строками и N столбцами (или булевых матриц $N \times N$) A и B — это тоже $N \times N$ матрица C , в которой элемент $C_{i,j}$, находящийся в строке i и столбце j , определяется следующим образом:

$$C_{i,j} = (A_{i,1} \wedge B_{1,j}) \vee (A_{i,2} \wedge B_{2,j}) \vee \dots \vee (A_{i,N} \wedge B_{N,j}),$$

где операция \wedge — это логическое умножение (или операция “И”), и операция \vee — это логическое сложение (или операция “ИЛИ”):

X	Y	$X \wedge Y$	$X \vee Y$
истина	истина	истина	истина
истина	ложь	ложь	истина
ложь	истина	ложь	истина
ложь	ложь	ложь	ложь

Умножение матриц в некоторых аспектах похоже на умножение чисел. Например, из предыдущего определения несложно показать, что умножение матриц ассоциативно: $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ для любых матриц A , B и C размером $N \times N$. В то же время есть и различия. Например, умножение матриц некоммукативно: может случиться, что $A \cdot B \neq B \cdot A$.

По аналогии с числами, возведение матриц в степень определяется через многократное последовательное умножение:

$$A^M = \underbrace{A \cdot A \cdot \dots \cdot A}_M$$

Например, в виде булевой матрицы A можно представить сеть улиц некоторого города. Элемент матрицы $A_{i,j}$ обозначает наличие улицы между площадями i и j . Матрицу A , возведённую в степень M , обозначим $X = A^M$. Тогда элемент $X_{i,j}$ показывает, что между площадями i и j существует маршрут, состоящий ровно из M отрезков улиц (то есть проходящий через $M - 1$ промежуточных площадей).

Написать программу, которая для данной $N \times N$ матрицы A вычисляет степень A^M .

Входные данные. В первой строке текстового файла `ma.sis` дана размерность матрицы N ($1 \leq N \leq 10$). В каждой из следующих N строк дано ровно N букв T (истина) или V (ложь): элементы матрицы A построчно. В последней строке файла дан показатель степени M ($1 \leq M \leq 1\,000\,000\,000$).

Выходные данные. В текстовый файл `ma.val` вывести ровно N строк, в каждую из них ровно N букв T или V: элементы матрицы A^M построчно.

Пример.

<code>ma.sis</code>	<code>ma.val</code>
3	VVT
VTV	TVV
VVT	VTV
TVV	
5	

1. Башенные часы

1 секунда

30 очков

Из окна мэра города видны башенные часы с четырьмя циферблатами, у которых часовые и минутные стрелки двигаются всегда с постоянной скоростью. Сами циферблаты прозрачные, и окно мэра находится в таком месте, что мэр видит два циферблата: один, обращённый к нему, и тот, что находится за ним (обращённый в противоположную сторону). Иногда случается так, что какие-то стрелки совпадают, и мэру кажется, что на двух циферблатах меньше чем 4 стрелки.

Написать программу, которая подсчитывает, сколько раз это событие происходит в заданном интервале времени.

Входные данные. В первой строке текстового файла `tk.sis` задано начало временного интервала в виде $TT:MM$ ($00 \leq TT \leq 23$, $00 \leq MM \leq 59$), и во второй строке — конец интервала в том же формате.

Выходные данные. В единственную строку текстового файла `tk.val` вывести целое число K , показывающее, сколько раз в заданном интервале времени совпадают как минимум две стрелки двух циферблатов. Начало и конец интервала считаются частью интервала (то есть если стрелки встретятся ровно в $TT:MM$, это событие тоже нужно посчитать).

Пример.	<code>tk.sis</code>	<code>tk.val</code>
	00:00	47
	12:00	

2. Встреча диссидентов

1 секунда

30 очков

В некоем тоталитарном государстве N городов, в каждом из которых есть некоторое количество диссидентов. Некоторые города соединены между собой дорогами. По случаю некоторых важных событий диссидентам надо собираться и обсуждать их между собой. Поскольку передвижения могут быть опасны для диссидентов, они хотят выбрать место для встречи так, чтобы суммарная длина пути всех диссидентов до места встречи была бы минимальная.

Написать программу, которая находит город, лучше всего подходящий для встречи.

Входные данные. В первой строке текстового файла `dk.sis` дано количество городов N ($2 \leq N \leq 100$) и количество дорог M ($1 \leq M \leq 1000$). Города пронумерованы $1 \dots N$. Во второй строке файла дано N разделённых пробелами целых чисел X_1, X_2, \dots, X_N , где X_i ($0 \leq X_i \leq 100$) обозначает количество диссидентов в городе номер i . В каждой из следующих M строк дано три целых числа A_i, B_i и L_i ($1 \leq A_i, B_i \leq N$, $0 < L_i \leq 1000$), которые означают, что между городами A_i и B_i проложена дорога длиной L_i . Известно, что между двумя городами не может быть более одной дороги, все дороги с двухсторонним движением, и из любого города можно добраться в любой другой.

Выходные данные. В единственную строку текстового файла `dk.val` вывести целые числа K и D , где K означает номер города, где должны встретиться диссиденты, и D означает суммарную длину пути всех диссидентов. Если оптимальных мест несколько, вывести любое из них.

Пример.

dk.sis	dk.val
4 3	3 4
1 1 1 2	
1 3 1	
2 3 1	
3 4 1	

3. Кубики

10 секунд

40 очков

На каждой грани кубика изображена одна цифра. При составлении числа из кубиков их кладут в ряд, и у каждого кубика оказывается видимой только одна грань.

Написать программу, которая находит комплект с минимальным количеством кубиков, из которого можно составить все числа $A \dots B$ (включая сами числа A и B).

Обратите внимание, что цифры 6 и 9 при переворачивании кубика превращаются друг в друга.

Входные данные. В единственной строке текстового файла `ak.sis` даны числа A и B ($0 \leq A \leq B \leq 750$).

Выходные данные. В первую строку текстового файла `ak.val` вывести необходимое количество кубиков N и в каждую из следующих N строк вывести ровно 6 цифр: описания кубиков, входящих в комплект. Цифры выводить рядом друг за другом, без пробелов. Поскольку цифры 6 и 9 замещаются друг другом, можно выводить любую из них. Если решений с минимальным количеством кубиков несколько, вывести любое из них.

Пример.

ak.sis	ak.val
10 20	2
	012345
	012678