

1. Последовательность

5 секунд

20 очков

Дана последовательность a_1, a_2, \dots, a_N , состоящая из N элементов. Значение каждого элемента указывает, к какому элементу можно следующим шагом перейти от этого элемента.

Написать программу, которая проверяет, можно ли, начав с первого элемента последовательности и совершая вышеуказанные шаги, вернуться опять к этому элементу.

Входные данные. На первой строке входного файла `JADA.SIS` дано число N ($1 \leq N \leq 100\,000$) — количество элементов последовательности. На второй строке — N разделённых пробелами целых числа — элементы последовательности a_1, a_2, \dots, a_N .

Выходные данные. В первой строке текстового файла `JADA.VAL` вывести слово `JAH`, если возможно вернуться к первому элементу, или слово `EI` в противном случае. В случае положительного результата вывести во второй строке минимальное необходимое количество шагов.

Пример.

	<code>JADA.SIS</code>	<code>JADA.VAL</code>
	4	JAH
	2 3 1 1	3

Замечание. Движение будет происходить следующим образом: $a_1 = 2 \rightarrow a_2 = 3 \rightarrow a_3 = 1 \rightarrow a_1$, всего 3 шага.

Пример.

	<code>JADA.SIS</code>	<code>JADA.VAL</code>
	4	EI
	2 3 2 1	

Замечание. Движение происходит следующим образом: $a_1 = 2 \rightarrow a_2 = 3 \rightarrow a_3 = 2 \rightarrow a_2 = 3 \rightarrow a_3 = 2 \rightarrow \dots$ и так далее до бесконечности.

Оценивание. За тесты с ответом `EI` получают очки лишь те программы, которые решат правильно хотя бы один тест с ответом `JAH`.

2. Родственники

5 секунд

40 очков

На праздник приглашено N гостей. Организаторам праздника известно, что некоторые из гостей родственники.

Конечно родственные отношения симметричны (т.е. если A родственник B , то B — родственник A) и транзитивны (т.е. если A — родственник B , а B — родственник C , то A — родственник C).

Написать программу, которая находит среди гостей группу людей максимального размера, такую что никакие два человека из этой группы не родственники.

Входные данные. На первой строке текстового файла `SUG.SIS` даны разделённые пробелом числа N и K ($1 \leq N \leq 100$, $0 \leq K \leq N(N-1)/2$), где N — число гостей, а K — число явно данных фактов о гостях. Гости пронумерованы числами от 1 до N . На каждой из следующих K строк находятся 2 разделённых пробелом числа A_i и B_i ($1 \leq A_i \leq N$, $1 \leq B_i \leq N$), которые указывают, что гости A_i и B_i родственники.

Выходные данные. В первой строке текстового файла `SUG.VAL` вывести размер найденной группы M и во второй строке вывести M разделённых пробелами чисел — членов группы. Если групп максимального размера несколько, вывести любую.

Пример.

	<code>SUG.SIS</code>	<code>SUG.VAL</code>
	5 3	2
	1 2	1 3
	3 4	
	5 1	

3. Файл crontab

открытые тесты

40 очков

В компьютерных системах часто надо выполнять некие действия регулярно по заданному графику. В UNIX-подобных системах для этого обычно используется программа `crond` (от греч. *chronos* ‘время’ и англ. *daemon* ‘дух’), которая считывает заданный график из файла `crontab`.

Каждая строка в файле `crontab` состоит из шести полей, разделённых символами пробел или табулятор. Пять полей (которые не должны содержать пробелы или табуляторы) описывают график выполнения какого-либо действия, шестое поле задаёт команду, которую необходимо запустить для выполнения этого действия.

График выполнения команды описывается следующими пятью полями: минуты (0...59), часы (0...23), дни (1...31), месяцы (1...12) и дни недели (0...6, где 0 = воскресенье, 1 = понедельник, ..., 6 = суббота). Каждое из пяти соответствующих полей в файле содержит одно или несколько значений. Несколько значений записываются в виде разделённой запятыми последовательности, состоящей из чисел или из пар чисел разделённых тире (-). Пара чисел *A-B* задаёт диапазон значений от *A* до *B* включая и *A* и *B*. Значением поля может также быть звёздочка (*). Звёздочка обозначает все возможные значения данного поля.

Команда запускается во время, которое соответствует значениям всех полей. К примеру, если в поле минут записано 0,30 в поле часов — 12-14, то команда будет запускаться в 12:00, 12:30, 13:00, 13:30, 14:00 и 14:30. Исключением являются поля ‘день’ и ‘день недели’. Если в обоих указана не звёздочка, то действие выполняется даже в случае если только одно из этих двух полей позволяет это. Например, если в поле ‘день’ записано 10,20 и в поле ‘день недели’ 5, то команда будет выполнена по 10. и 20. числам месяца (независимо от дня недели) и каждую пятницу (независимо от числа).

Дан список команд, которые были запущены программой `crond` за некий промежуток времени. Указаны времена, когда была запущена каждая команда. Необходимо создать файл `crontab` который мог бы быть причиной такой работы `crond` в этот промежуток времени. Желательно, чтобы количество строк в созданном файле было по возможности минимально. Если несколько команд должны были запущены одновременно, на самом деле их запустили в порядке файла `crontab`. Создаваемый файл `crontab` должен сохранить этот порядок.

Входные данные. Каждая строка файла `CRON.SIS` содержит три разделённых пробелами значения: дата, когда была запущена команда в формате ДД.ММ.ГГГГ, время в формате ЧЧ:ММ и выполненная команда. Известно, что файл содержит записи о всех запущенных командах в промежутке между временами, указанными на первой и последней строках файла. Строки в файле упорядочены в хронологическом порядке.

Выходные данные. В файл `CRON.VAL` вывести `crontab`, при использовании которого `crond` запускал бы команды в указанный период времени точно так, как это указано во входном файле. Поведение `crond` до момента, указанного на первой строке входного файла, и после момента, указанного на последней строке входного файла, несущественно.

Пример.	CRON.SIS	CRON.VAL
	20.10.2003 12:00 test	0 12-14,17 * * * test
	20.10.2003 13:00 test	
	20.10.2003 14:00 test	
	20.10.2003 17:00 test	
	21.10.2003 12:00 test	

Замечание. Описанный в этом выходном файле `crontab` запускает команду `test` каждый день в 12:00, 13:00, 14:00 и 17:00. В промежутке от 20.10.2003 12:00 до 21.10.2003 12:00 команда будет запускаться точно в те моменты, которые были указаны во входном файле.

Поэтому этот `crontab` удовлетворяет условиям задачи несмотря на то, что команда `test` запускается и до 20.10.2003 12:00 и после 21.10.2003 12:00, в частности 21.10.2003 в 13:00, 14:00 и 17:00.

Оценивание. Имеется 10 входных файлов с именами от `cron01.sis` до `cron10.sis`. В качестве решения необходимо представить соответствующие выходные файлы с именами от `cron01.val` до `cron10.val`. Программу, использованную для решения представлять необязательно. Она не оценивается. За каждый предоставленный в качестве решения корректный выходной файл участник получает количество очков, обратно пропорциональное отношению количества строк в представленном файле к количеству строк в лучшем решении (т.е. файл содержащий в два раза больше строк получит в два раза меньше очков).