

1. Среднее арифметическое

1 секунда

10 очков

Написать программу, которая проверит, является ли какое-либо из заданных четырёх целых чисел средним арифметическим остальных.

Входные данные. В единственной строке текстового файла `kesk.sis` дано четыре разделённых пробелами целых числа (их значения лежат в промежутке $-30\,000 \dots 30\,000$).

Выходные данные. В первую строку текстового файла `kesk.val` вывести слово `EI`, если ни одно из заданных четырёх чисел не является средним арифметическим всех остальных. Если такое число найдено, то вывести в первую строку слово `JAH` и во вторую строку — это число.

Пример.

<code>kesk.sis</code>	<code>kesk.val</code>
1 2 3 4	EI

Пример.

<code>kesk.sis</code>	<code>kesk.val</code>
7 2 3 4	JAH
	4

Оценивание. В этом задании за тесты с ответом `EI` очки получают только те программы, которые правильно решат хотя бы один тест с ответом `JAH`.

2. Часы

1 секунда

20 очков

Написать программу, которая находит угол между стрелками часов в заданный момент времени.

Входные данные. В единственной строке текстового файла `kell.sis` задан момент времени в виде `HH : MM`, где `HH` — это часы, а `MM` — это минуты (оба — двухзначные числа).

Выходные данные. В единственную строку текстового файла `kell.val` вывести угол α (в градусах) между часовой и минутной стрелками обычных часов в момент `HH : MM`. Выводимое значение должно удовлетворять неравенству $0 \leq \alpha \leq 180$ и не должно отличаться от правильного ответа больше, чем на 0,01 градус.

Пример.

<code>kell.sis</code>	<code>kell.val</code>
03:00	90

Пример.

<code>kell.sis</code>	<code>kell.val</code>
15:30	75

3. Тройка

1 секунда

30 очков

Дано три числовых последовательности, упорядоченных в неубывающем порядке.

Написать программу, которая из каждой последовательности выберет по одному числу, так чтобы диаметр этой тройки был минимальным. Диаметром тройки чисел называется разница между максимальным и минимальным значениями в этой тройке.

Входные данные. В первой строке текстового файла `kolm.sis` дана длина последовательностей N ($1 \leq N \leq 10\,000$). Во второй строке файла дано N разделённых пробелами целых числа A_1, A_2, \dots, A_N ($-10\,000 \leq A_1 \leq A_2 \leq \dots \leq A_N \leq 10\,000$) — элементы первой последовательности. Аналогично, в третьей строке файла задана последовательность B_1, B_2, \dots, B_N , и в четвёртой строке — последовательность C_1, C_2, \dots, C_N .

Выходные данные. В единственную строку текстового файла `kolm.val` вывести три разделённых пробелами целых числа I, J, K , таких что диаметр тройки чисел A_I, B_J, C_K и является минимальным. Если троек с минимальным диаметром несколько, вывести любую из них.

Пример.	<code>kolm.sis</code>	<code>kolm.val</code>
	4	3 1 3
	0 1 2 5	
	3 4 6 7	
	0 1 2 8	

Примечание. Полный перебор всех возможных троек не успеет отработать за секунду в случае достаточно длинных последовательностей. Чтобы получить положительный результат во всех тестах, нужно придумать и реализовать более эффективный алгоритм. Среди примерах входных и выходных файлов имеется входной файл максимального размера.

4. Рекламный плакат

1 секунда

40 очков

Рекламная фирма хочет наклеить плакат на забор. Но к сожалению, забор сделан не совсем корректно. Все доски шириной в пядь и поставлены вертикально рядом друг с другом на ровной поверхности, но они далеко не одинаковой высоты.

Рекламная фирма согласна изготовить плакат любой высоты и ширины, но он должен быть прямоугольной формы и не должен быть выше, чем край забора.

Написать программу, которая для данного забора находит максимальную возможную площадь плаката.

Входные данные. В первой строке текстового файла `rekl.sis` дано число досок забора N ($1 \leq N \leq 50\,000$), и во второй строке дано N разделённых пробелами целых чисел A_i ($0 \leq A_i \leq 100$), где A_i означает высоту i -ой доски в пядях, начиная от основания забора.

Выходные данные. В единственную строку текстового файла `rekl.val` вывести максимальную площадь плаката (в квадратных пядях), который можно наклеить на данный забор.

Пример.	<code>rekl.sis</code>	<code>rekl.val</code>
	4	9
	1 3 3 4	

Максимальную площадь получим, если наклеим плакат размером 3×3 , начиная со второй доски.

5. Протокол финиша

1 секунда

50 очков

В лыжных соревнованиях каждый спортсмен на финише получает флажок, на котором написано его место в текущем рейтинге в формате A/B , где B означает количество лыжников, уже финишировавших к этому моменту, а число A показывает место участника среди них. Спортсмен, который первым прибывает на финиш, всегда получает флажок $1/1$, второй получает $1/2$ или $2/2$, третий — $1/3$, $2/3$ или $3/3$ и т.д.

Написать программу, которая получает данные обо всех флажках и находит место каждого спортсмена в конечном рейтинге. Можно считать, что “деления” мест нет, т.е. никакие два лыжника не заняли одинаковые места.

Входные данные. В первой строке текстового файла `prot.sis` дано количество спортсменов N ($1 \leq N \leq 50\,000$), и в каждой из следующих N строк дано описание флажка участника номер i в виде A_i/B_i ($1 \leq A_i \leq B_i \leq N$). Флажки могут встречаться в файле в любом порядке.

Выходные данные. В текстовый файл `prot.val` вывести ровно N строк, в каждую строку по числу. В строку номер i вывести конечное место того участника, чьи данные были заданы во входном файле в $i + 1$ -ой строке.

Пример.	<code>prot.sis</code>	<code>prot.val</code>
	5	5
	1/1	1
	1/2	3
	2/3	4
	3/4	2
	2/5	

Обозначаем участников А...Е в порядке их прибытия на финиш. А получает флажок $1/1$. В получает флажок $1/2$, следовательно, в этот момент рейтинг такой: В, А. Следующим идёт С, он получает флажок $2/3$, следовательно, новый рейтинг: В, С, А. Потом D получает флажок $3/4$, следовательно, новый рейтинг: В, С, D, А. Наконец, участник Е получает флажок $2/5$, следовательно, финальный рейтинг: В, Е, С, D, А.

6. Крыса

50 очков

Крысу пускают в трёхмерный лабиринт, в котором каждый переход соединяет между собой две камеры. В каждом переходе есть дверь на пружине, которая позволяет крысе проходить этот проход только в одном направлении. При этом все ходы расположены таким образом, что крыса, выйдя из какой-либо камеры, больше не может в неё вернуться.

В одной из камер есть кусок сыра. Написать программу, которая подсчитывает, сколько различных путей есть у крысы от её начального положения до камеры с сыром. Два пути считаются одинаковыми, если они проходят через одни и те же камеры в том же порядке.

Входные данные. В первой строке текстового файла дано количество камер N ($1 \leq N \leq 1000$) и количество ходов M ($1 \leq M \leq \frac{N \cdot (N-1)}{2}$). Камеры пронумерованы числами $1 \dots N$, причём крыса начинает движение из камеры номер 1, а сыр находится в камере номер N . В каждой из следующих M строк дано два разделённых пробелами целых числа A_i и B_i ($1 \leq A_i \leq N$, $1 \leq B_i \leq N$, $A_i \neq B_i$), которые означают, что в лабиринте есть ход из камеры A_i в камеру B_i .

Выходные данные. В единственную строку текстового файла вывести одно целое число — количество различных маршрутов из начальной камеры в камеру с сыром.

Пример.	<code>rott.sis</code>	<code>rott.val</code>
	5 6	4
	1 2	
	2 3	
	1 3	
	3 4	
	4 5	
	3 5	

Примечание. Ответ не обязательно поместится в 32- или 64-битную переменную.

Оценивание. В этом задании дано 10 комплектов входных данных в файлах с именами от `rotttest.01.sis` до `rotttest.10.sis`, и в качестве решения необходимо предоставить соответствующие комплекты выходных данных в файлах с именами от `rotttest.01.val` до `rotttest.10.val`. Представлять саму программу не обязательно, она не оценивается.