

1. Компьютерный магазин

1 секунда

30 очков

В компьютерный магазин часто приходят клиенты, желающие купить компьютер определённой цены или мощности. Написать программу, помогающую продавцам найти среди имеющихся компьютеров все, которые отвечают требованиям клиента.

Входные данные. В первой строке текстового файла `apood.sis` дано число компьютеров N ($0 \leq N \leq 10\,000$), а в каждой из N следующих описание одного компьютера. Описания имеют вид $H P M K V$, где H цена компьютера, P скорость процессора, M объём RAM, K объём жёсткого диска и V объём видео RAM. Значения всех параметров целые числа $0 \dots 10\,000$.

В последней строке файла условие вида $X Y Z$, где X символ из множества $\{H, P, M, K, V\}$, Y оператор сравнения $=$ или $>$ и Z целое число ($0 \leq Z \leq 10\,000$).

Выходные данные. В первую строку текстового файла `apood.val` вывести число удовлетворяющих условию компьютеров R , а в следующие R строки описания этих компьютеров в порядке задания их во входном файле. Если ни один компьютер не устраивает клиента, вывести в единственную строку файла сообщение `EI OLE`.

Пример.	<code>apood.sis</code>	<code>apood.val</code>
	4	2
	7000 1600 256 80 64	5000 1200 128 60 32
	5000 1200 128 60 32	2000 400 64 30 16
	10000 2400 512 120 128	
	2000 400 64 30 16	
	H =< 6000	

Пример.	<code>apood.sis</code>	<code>apood.val</code>
	3	EI OLE
	400 600 128 2 16	
	100 120 64 1 8	
	700 800 190 10 32	
	M >= 256	

Оценивание. В этом задании за негативные тесты (с ответом `EI OLE`) получают очки только те программы, которые успешно пройдут хотя бы один положительный тест.

2. Замки

1 секунда

30 очков

Искатель кладов оказался в пещере, единственный выход которого закрыт несколькими замками. Чтобы выйти из пещеры, надо иметь ключи от всех замков. К счастью, искатель кладов имеет с собой инструкции для выхода из пещеры. К несчастью, он не знает, правильные ли они. . .

Написать программу, которая на базе карты пещеры и этих инструкции решит, может ли он собрать все ключи и дойти после этого к выходу пещеры.

На каждом шагу искатель кладов может либо двигаться на одну клетку карты вперёд или назад, либо повернуться на 90° налево или направо. В начальном положении он стоит лицом на север (на карте сверху). При проходе по лабиринте любое доступное место (в том числе место нахождения выхода) можно посетить несколько раз, но пройти через стены или выйти за пределы карты нельзя.

Входные данные. В первой строке текстового файла `lukud.sis` размеры карты N и M ($1 \leq N, M \leq 20$). В каждой из следующих N строк ровно M символов: карта пещеры, на которой 0 обозначает проход, 1 стену, K место нахождения ключа, S исходную позицию искателя кладов и F место нахождения выхода. Известно, что все ключи находятся в проходах (не в стенах) и их не больше 50.

В следующей строке файла целое L ($0 \leq L \leq 250$) и в следующей за ней строке ровно L знаков: инструкции искателя кладов, где F и B обозначают движение вперёд и назад, а L и R повороты на 90° налево и направо.

Выходные данные. В первую строку текстового файла `lukud.val` вывести слово `EI`, если искатель

кладов не может выйти из пещеры, слово VIGA при попытке проходить через стену или выходить за пределы карты, и слово JAH, если он может выйти. В двух последних случаях вывести во вторую строку номер шага, на которой это событие произойдет ($1 \dots L$).

Пример.

lukud.sis	lukud.val
3 3	EI
1K0	
F1S	
000	
6	
BLFFRF	

Искатель кладов доидёт до выхода, но не имеет с собой ключ, необходимый для открытия двери.

Пример.

lukud.sis	lukud.val
3 3	VIGA
1K0	2
F1S	
000	
5	
LFBLF	

На втором шагу искатель кладов сталкивается в стену.

Пример.

lukud.sis	lukud.val
3 3	JAH
1K0	12
F1S	
000	
13	
FLFBLFFRFFRFF	

На последнем шагу искатель кладов сталкивался бы в стену, но это не имеет значения так как он до этого уже вышел из пещеры.

3. Сумма

1 секунда 40 очков

Даны N целых чисел X_1, X_2, \dots, X_N . Написать программу, которая расставит между ними знаки плюс и минус так, чтобы значение получившегося выражения было равно заданному целому S .

Входные данные. В первой строке текстового файла `summa.sis` длина цепочки N ($2 \leq N \leq 24$), во второй строке её элементы X_i ($0 \leq X_i \leq 50\,000\,000$, $1 \leq i \leq N$), а в третьей искомая "сумма" S ($-1\,000\,000\,000 \leq S \leq 1\,000\,000\,000$).

Выходные данные. В единственную строку текстового файла `summa.val` вывести найденное равенство, или EI OLE, если получить требуемый результат невозможно. Если решение не единственное, вывести любое.

Пример.

summa.sis	summa.val
3	15+25-30=10
15 25 30	
10	

Пример.

summa.sis	summa.val
2	EI OLE
10 10	
100	

Оценивание. В этом задании за негативные тесты (с ответом EI OLE) получают очки только те программы, которые успешно пройдут хотя бы один положительный тест.