

1. ПРЯМОУГОЛЬНИКИ

10 очков

10 секунд

На плоскости даны три прямоугольника, стороны которых параллельны координатным осям. Написать программу, которая находит площадь их пересечения.

Входные данные. В текстовом файле `RIST.SIS` три строки, каждая из них содержит в виде $x_1 y_1 x_2 y_2$ координаты левой нижней и правой верхней вершин одного прямоугольника, где все координаты – целые числа, абсолютное значение которых не превышает 1000.

Выходные данные. В текстовый файл `RIST.VAL` вывести одно целое число – площадь пересечения прямоугольников.

Пример.

<code>RIST.SIS</code>	<code>RIST.VAL</code>
10 10 70 40	400
20 20 80 50	
30 30 90 60	

2. СУММА

20 очков

10 секунд

Дана таблица состоящая из целых чисел. Написать программу для нахождения наибольшей возможной суммы на пути из нижнего левого в правый верхний угол, причём на каждом шагу можно двигаться на одну клетку вверх или направо, а суммировать следует значения всех пройденных клеток.

1	2	3
7	6	3
-1	4	5
3	5	4

Таблица

1	2	3
7	6	3
-1	4	5
3	5	4

Сумма 15

1	2	3
7	6	3
-1	4	5
3	5	4

Сумма 24

Входные данные. На первой строке текстового файла `SUMMA.SIS` записано два целых числа – высота таблицы n ($1 \leq n \leq 100$) и ширина m ($1 \leq m \leq 100$). На следующих n строках записано по m целых чисел, чьё абсолютное значение не превышает 1000 – соответствующая строка таблицы.

Выходные данные. В текстовый файл `SUMMA.VAL` вывести искомую максимальную сумму.

Пример.

<code>SUMMA.SIS</code>	<code>SUMMA.VAL</code>
4 3	24
1 2 3	
7 6 3	
-1 4 5	
3 5 4	

3. ФЕРЗИ

30 очков

25 секунд

Написать программу, которая располагает K ферзей на шахматной доске размером $N \times M$ так, чтобы ни один из них не был под ударом другого (ферзь может ходить по горизонтали, вертикали, и диагоналям).

Входные данные. На первой строке текстового файла `LIPUD.SIS` записаны высота шахматной доски N ($1 \leq N \leq 25$) и ширина M ($1 \leq M \leq 25$) и на второй строке число ферзей K ($1 \leq K \leq 25$).

Выходные данные. В текстовый файл `LIPUD.VAL` вывести в точности K строк – на каждую вывести разделённые пробелом номер строки ($1..N$) и столбца ($1..M$) одного из ферзей. Порядок ферзей в файле не имеет значения. Если решений много, вывести любое из них. Входные данные таковы, что решение всегда найдётся.

<u>Пример.</u>	<code>LIPUD.SIS</code>	<code>LIPUD.VAL</code>
	3 4	1 1
	3	3 2
		2 4

4. НЕРАВЕНТЦВА

35 очков

10 секунд

Дано множество неравенств вида $x < y$ или $x > y$. Написать программу, которая находит такую последовательность присутствующих в неравенствах переменных, которая удовлетворяет все неравенства.

Входные данные. На первой строке текстового файла `VORRAT.SIS` записано количество неравенств N ($1 \leq N \leq 1000$) и на следующих N строках записано по одному неравенству в виде $x < y$ или $x > y$, где x и y – однобуквенные имена переменных. Большие и маленькие буквы считать различными.

Выходные данные. На первую строку текстового файла `VORRAT.VAL` вывести количество присутствующих в неравенствах переменных K и на следующие K строк имена переменных в порядке, удовлетворяющем все неравенства, причём наименьшая переменная должна идти первой. Если решений много, вывести любое из них. Входные данные таковы, что решение всегда найдётся.

<u>Пример.</u>	<code>VORRAT.SIS</code>	<code>VORRAT.VAL</code>
	3	4
	a<b	a
	b<c	b
	d>c	c
		d

5. IPv4

35 очков

10 секунд

В соответствии с протоколом Internet Protocol version 4 (IPv4) адрес устройства подключённого к сети есть 32-битное число без знака, которое можно записать в одном из следующих текстовых представлений: $a.b.c.d$, $a.b.c$, $a.b$ или a . В представлении $a.b.c.d$ каждая из частей должна быть 8-битным целым числом. В представлении $a.b.c$ части a и b должны быть 8-битными, а часть c 16-битной. В представлении $a.b$ часть a должна быть 8-битной, а часть b 24-битной. В последнем из представлений часть a должна быть 32-битной.

Каждая часть адреса может быть представлена в системе исчисления по основанию 4, 8, 10 или 16. Если часть начинается с последовательности $0x$ (ноль и буква икс), то последующие символы описывают число в 16-ричной системе. Если последовательность начинается с 0 (ноль), то в 8-ичной. Если последовательность начинается с $0!$ (ноль и восклицательный знак), то в 4-ичной. В остальных случаях используется 10-ичная система.

Например представление $a.b.c.d$ адреса $11000001001010000000010101111100$ в 10-ичной системе есть $193.40.5.124$, а представление $a.b$ в 16-ричной системе есть $0xc1.0x28057c$. Также этот адрес можно записать в виде $0301.40.0x57c$. А например представление $301.40.5.124$ не является корректным, так как её первая часть (число 301 в 10-ичной системе или 100101101 в 2-ичной) не умещается в 8 битов.

Написать программу, которая опознаёт корректные представления адресов.

Входные данные. На первой строке текстового файла `IPV4.SIS` записано число n ($1 \leq n \leq 100$) и на следующих n строках по последовательности символов длиной не более 100 знаков. В 16-ричной системе большие и маленькие буквы не различать.

Выходные данные. В текстовый файл `IPV4.VAL` вывести в точности n строк – по одной строке на каждую последовательность символов. Если последовательность есть корректное представление адреса IPv4, вывести этот адрес в виде 32-местного 2-ичного числа (вместе с начальными нулями), в обратном случае вывести `VIGA`.

<u>Пример.</u>	<code>IPV4.SIS</code>	<code>IPV4.VAL</code>
	5	11000011010100000110000011011110
	195.80.96.222	VIGA
	0x200.01	00011100010101010001001000010010
	0x1C551212	10101011110000010010001101000101
	0XAb.0xc12345	VIGA
	ue831oue0q3	

Замечание. На самом деле в IPv4 4-ричной системой не пользуются. Эта система была добавлена в задание, чтобы сделать его интереснее.