

1. Калькулятор

1 секунда 30 очков

В позиционной системе счисления с основанием b (т.н. b -ичной системе) числа записываются с помощью цифр $0 \dots b-1$ и последовательность цифр $a_n a_{n-1} \dots a_1 a_0$ задаёт число $a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + \dots + a_1 \cdot b + a_0$. Если $b > 10$, то в дополнение к обычным арабским цифрам в роли цифр используются латинские буквы.

К примеру цифрами в 16-чной системе являются $0 \dots 9$ и $A \dots F$ ($=10 \dots 15$). Значением числа $A001$ в 16-чной системе будет $10 \cdot 16^3 + 0 \cdot 16^2 + 0 \cdot 16 + 1 = 40961$.

Реализовать программируемый калькулятор для преобразования чисел из одной системы счисления в другую.

Входные данные. На первой строке текстового файла `ekalk.sis` длина программы N ($1 \leq N \leq 1000$), а на каждой из следующих N строк по одной команде. Команды бывают трёх типов:

- `ibase= B_1` поменять входную систему: в дальнейшем все числа в программе заданы в B_1 -чной системе ($2 \leq B_1 \leq 36$);
- `obase= B_2` поменять выходную систему: все числа далее в программе будут выводиться в B_2 -чной системе ($2 \leq B_2 \leq 36$);
- `A` вывести число A в текущей системе вывода ($0 \leq A \leq 1\,000\,000\,000$).
(Все числа в этом параграфе даны в десятичной системе.)

В начале работы программы входной и выходной системой является десятичная.

Выходные данные. В текстовом файле `ekalk.val` записать вывод калькулятора после исполнения данной программы. В выводе будет по одной строчке для каждой команды третьего типа. Во входном файле будет по крайней мере одна такая команда.

Пример.	<code>ekalk.sis</code>	<code>ekalk.val</code>
	5	40961
	<code>ibase=16</code>	A001
	A001	
	<code>obase=10</code>	
	<code>ibase=A</code>	
	40961	

2. Многоугольник

1 секунда 30 очков

На плоскости даны N точек. Требуется найти какой-либо многоугольник с вершинами в данных точках.

Примечание. Многоугольник — это замкнутая ломаная, не пересекающая саму себя.

Входные данные. На первой строке текстового файла `enurk.sis` дано количество точек N ($3 \leq N \leq 1000$) и на следующих N строках — координаты точек. Координаты даны в виде пар целых чисел по абсолютному значению не превышающих 10 000. Точки пронумерованы в порядке их следования в файле числами $1 \dots N$.

Выходные данные. На единственной строки текстового файла `enurk.val` вывести N разделённых пробелами чисел — номера точек в порядке их следования в многоугольнике. Можно предполагать, что в каждом тесте найдётся по крайней мере одно решение.

Пример.	<code>enurk.sis</code>	<code>enurk.val</code>
	4	1 2 4 3
	2 1	
	9 2	
	3 7	
	8 9	

3. Е-ритметика

1 секунда

40 очков

Е-ритметика — это компьютерная игра для одного человека, в которую играют на поле $N \times M$ клеток. В начале игры все клетки пусты. С каждым ходом игрок щёлкает мышью на одной из пустых клеток, в результате чего

1. в клетку, на которой щёлкнули, записывается число 1;
2. соседние пустые клетки остаются пустыми;
3. числа в соседних непустых клетках увеличиваются на 1.

Две клетки считаются соседними, если у них одно общее ребро.

Цель игрока — добиться на поле определённой конфигурации. Написать программу, которая поможет это сделать.

Входные данные. На первой строке текстового файла `eritm.sis` дано количество рядов N и столбцов M игрового поля ($1 \leq N \leq 100$, $1 \leq M \leq 100$). На каждой из следующих N строк даны M разделённых пробелами целых чисел $1 \dots 5$: конфигурация поля, которую необходимо получить. Ряды на поле пронумерованы сверху вниз числами $1 \dots N$, а столбцы — $1 \dots M$ слева направо.

Выходные данные. На первой строке текстового файла `eritm.val` вывести количество ходов K , за которое можно получить требуемое поле, и на следующих K строках по два разделённых пробелом целых числа: номер ряда и столбца клетки, на которой следует щёлкнуть в соответствующий ход. Щёлкать можно только на пустых клетках. Если требуемой конфигурации можно добиться несколькими способами, вывести любой из них. Если же решения не существует, вывести на единственной строке слова `EI SAA`.

Пример.	<code>eritm.sis</code>	<code>eritm.val</code>
	2 3	6
	3 3 1	1 1
	1 2 3	1 2
		2 3
		1 3
		2 2
		2 1

Пример.	<code>eritm.sis</code>	<code>eritm.val</code>
	2 2	EI SAA
	1 1	
	1 1	

Оценивание. Очки получают лишь те решения, которые пройдут хотя бы один тест в котором решение найдётся.