

1. Треугольник Паскаля

5 секунд

10 очков

Треугольник Паскаля — бесконечная треугольная таблица, в каждой строке которой на один элемент больше, чем в предыдущей. Первый и последний элементы каждой строки равны 1, любой другой элемент находится как сумма двух находящихся над ним элементов. Например, изображённый на рисунке жирным шрифтом элемент **3** равен сумме находящихся над ним элементов **1** и **2**. Написать программу, которая вычисляет указанный элемент треугольника Паскаля.

		1		
		1	1	
	1	2	1	
1	3	3	1	

Входные данные. В единственной строке текстового файла `KOLM.SIS` находится номер строки R искомого элемента и его позиция P в этой строке ($0 \leq P \leq R \leq 30$). Строки треугольника пронумерованы сверху вниз, начиная с нуля, позиции в строках пронумерованы слева направо, также начиная с нуля.

Выходные данные. В единственную строку текстового файла `KOLM.VAL` вывести значение указанного элемента.

Пример.

<code>KOLM.SIS</code>	<code>KOLM.VAL</code>
<code>3 1</code>	<code>3</code>

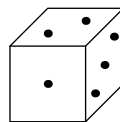
2. Катание игральной кости

5 секунд

20 очков

Прямоугольное игровое поле состоит из единичных квадратов. В каждой клетке поля находится одно число $1 \dots 6$. В некоторой клетке поля лежит игральная кость (единичный куб). Затем игральную кость начинают катать так, что на каждом шаге кость перекачивается без проскальзывания из своей клетки в одну из соседних. При этом требуется, чтобы число в клетке всегда совпадало с числом точек на нижней стороне кости, то есть той, которая касается стола. Выходить за пределы игрового поля запрещается. Написать программу, которая проверяет, соответствует ли данная последовательность вращений описанным требованиям.

Кость стандартная “правосторонняя”: суммарное количество точек на любых двух противоположных гранях равно 7, и в вершине, где пересекаются грани 1, 2 и 3, эти грани смотрятся по часовой стрелке именно в таком порядке:



Входные данные. В первой строке текстового файла `VEER.SIS` находятся количество рядов R и количество столбцов V ($1 \leq R \leq 30, 1 \leq V \leq 30$) игрового стола. В каждой из следующих R строк находится ровно V чисел $1 \dots 6$. В следующей строке дано исходное положение кости: номер ряда r_0 , номер столбца v_0 и количество точек на северной грани кости. В следующей строке находится длина последовательности перекачиваний N ($1 \leq N \leq 50$), и в последней строке ровно N букв `N`, `E`, `S`, `W`, которые обозначают перекачивания кости соответственно в северном, восточном, южном и западном направлениях. Ряды игрового стола пронумерованы от севера к югу $1 \dots R$ и столбцы от запада к востоку $1 \dots V$. Начальное положение кости отвечает всем требованиям.

Выходные данные. В первую строку текстового файла `VEER.VAL` вывести слово `JAH`, если заданная последовательность перекачиваний соответствует требованиям, и слово `EI`, если не соответствует. В последнем случае вывести также во вторую строку файла номер хода, на котором впервые нарушаются правила.

Пример.	VEER.SIS	VEER.VAL
	2 3	EI
	1 2 6	4
	5 3 4	
	1 1 3	
	5	
	EESWW	

Оценивание. В этом задании за JAH-тесты очки получают только те программы, которые пройдут хотя бы один EI-тест.

3. Мозаика

5 секунд

30 очков

Из единичных квадратов, края которых окрашены, необходимо собрать прямоугольную мозаику, в которой два кусочка могут находиться рядом, только если они соприкасаются сторонами, окрашенными в один цвет. Написать программу, которая составляет из данного набора кусочков мозаику, соответствующую описанным требованиям.

Входные данные. В первой строке текстового файла MOSA.SIS находится число цветов C ($1 \leq C \leq 5$) в мозаике, во второй строке дано число рядов R и число столбцов V искомой мозаики ($1 \leq R \leq 10$, $1 \leq V \leq 10$). В каждой из следующих $R \times V$ строк дано описание одного кусочка: цвет верхнего, правого, нижнего и левого края $c_{i,j}$ ($0 \leq c_{i,j} \leq C$), где 0 означает неокрашенный край, который должен быть на краю мозаики. Кусочки необходимо класть в мозаику именно в том положении, в каком они описаны, поворачивать их запрещено.

Выходные данные. В текстовый файл MOSA.VAL вывести ровно R строк, в каждую строку ровно V разделённых пробелами чисел — описание мозаики, где каждое число означает порядковый номер кусочка, положенного в соответствующее место мозаики. Кусочки пронумерованы числами $1 \dots R \times V$ в том порядке, в каком они даны во входном файле. Если решений несколько, вывести любое из них. Можно предполагать, что решение всегда найдётся.

Пример.	MOSA.SIS	MOSA.VAL
	4	2 3
	2 2	4 1
	2 0 0 3	
	0 1 4 0	
	0 0 2 1	
	4 3 0 0	

4. Окно

5 секунд

40 очков

Пользовательский интерфейс программы состоит из основного окна и находящихся поверх него вспомогательных окон, которые могут закрывать часть основного окна. Написать программу, которая на основании координат всех окон вычисляет, какая часть основного окна видна на экране.

Входные данные. В первой строке текстового файла AKEN.SIS описано положение основного окна, во второй строке находится количество вспомогательных окон N ($1 \leq N \leq 100$), и в каждой из следующих N строк — положение одного вспомогательного окна. Положения всех окон даны как целочисленные координаты верхнего левого и правого нижнего углов: $x_1 y_1 x_2 y_2$ ($0 \leq x_1 \leq x_2 < 800$, $0 \leq y_1 \leq y_2 < 600$). Вспомогательные окна не обязательно находятся полностью над основным окном.

Выходные данные. В единственную строку текстового файла AKEN.VAL вывести площадь в пикселях той части основного окна, которая не закрыта вспомогательными окнами.

Пример.	AKEN.SIS	AKEN.VAL
	0 0 50 25	930
	2	
	10 5 30 15	
	20 10 40 20	