

1. ЗАПРЕЩЁННОЕ СЛОВО

100 очков

10 секунд

Написать программу, которая подсчитывает сколько разных слов заданной длины можно составить из заданных букв так, чтобы составленное слово не содержало заданное подслово. Слово S_1 называют подсловом слова S_2 , если слово S_1 можно получить из слова S_2 путём удаления некоторого количества букв из начала и конца.

Входные данные. На первой строке файла KEELAT.SIS расположены заданные буквы (попарно различные маленькие и большие латинские буквы), на второй строке расположена длина слова N ($1 \leq N \leq 200$) и на третьей строке запрещённое подслово S (состоящее из заданных букв, длиной не более N).

Выходные данные. На единственную строку файла KEELAT.VAL вывести количество слов длины N состоящих из заданных букв, которые не содержат подслово S . Заданные буквы можно использовать повторно. Маленькие и большие буквы считать различными.

Пример.

KEELAT.SIS	KEELAT.VAL
ABC	4
2	
B	

2. ТАБЛИЦА

100 очков

15 секунд

Дана таблица целых чисел. Назовём весом прямоугольника состоящего из элементов таблицы сумму входящих в него чисел. Написать программу, которая находит разбиение заданной таблицы на прямоугольники, при котором вес прямоугольника с наибольшим весом был бы наименьшим.

Входные данные. На первой строку файла PAANID.SIS записано количество рядов R ($1 \leq R \leq 15$) и столбцов V ($1 \leq V \leq 15$) в таблице. На следующих R строках записаны элементы таблицы сверху вниз и слева направо: на каждой строке V целых чисел, абсолютное значение которых не превышает 1000.

Выходные данные. На первую строку файла PAANID.VAL вывести количество прямоугольников N в разбиении и на следующие N строк вывести описания прямоугольников в виде $r_1 \ v_1 \ r_2 \ v_2$, где $r_1 \ v_1$ – координаты левого верхнего а $r_2 \ v_2$ – координаты правого нижнего элементов. Найденное разбиение должно покрывать всю таблицу, прямоугольники не должны между собой пересекаться и наибольший вес прямоугольника должен быть минимальным. Порядок прямоугольников в файле не имеет значения.

Пример.

PAANID.SIS	PAANID.VAL
2 3	3
2 -10 5	1 1 2 2
10 2 6	1 3 1 3
	2 3 2 3

3. ПРЯМОУГОЛЬНИКИ

100 очков

10 секунд

На плоскости дано множество четырёхугольников. Написать программу, которая находит среди них прямоугольники и составляет из них последовательность максимальной длины, где каждый следующий прямоугольник помещается в предыдущий. Прямоугольник R_1 помещается в прямоугольник R_2 , если R_1 и R_2 можно так расположить относительно друг-друга, что ни одна точка R_1 не находится на границах R_2 или вне его.

Входные данные. На первой строке файла RISTK.SIS записано количество четырёхугольников N ($1 \leq N \leq 200$) и на следующих N строках на каждой x и y координаты вершин одного четырёхугольника записанные по направлению часовой стрелки. Все координаты – целые числа, чьё абсолютное значение не превышает 1000. Четырёхугольники пронумерованы $1..N$ в порядке их присутствия в файле.

Выходные данные. На первую строку RISTK.VAL вывести длину найденной последовательности K и на следующие K строк порядковые номера четырёхугольников в найденной последовательности. Если решений много, то вывести любое из них.

<u>Пример.</u>	RISTK.SIS	RISTK.VAL
	4	3
	120 10 120 70 220 70 220 10	1
	0 0 0 60 5 60 5 0	4
	110 170 140 130 100 100 70 140	2
	10 20 10 45 85 45 85 20	