

1. Линейные шашки

1 секунда 20 очков

Линейные шашки — это настольная игра, которую играют на поле $1 \times N$ клеток. В отличие от обычных шашек фишки в линейных шашках все одного цвета, но могут быть разной высоты. Состояние игры задаётся последовательностью a_1, a_2, \dots, a_N , где a_i даёт высоту фишки в клетке i (если там нет фишки, то $a_i = 0$).

Из клетки i можно сделать ход в клетку k только в том случае, если $i < k$ и все фишки в клетках $i + 1 \dots k$ ниже фишки в клетке i (поле k не обязано быть пустым). Разницу $k - i$ назовём длиной хода $i \rightarrow k$. Написать программу для определения самого длинного хода в заданной позиции.

Входные данные. На первой строке текстового файла `RK.SIS` дана длина игрового поля N ($1 \leq N \leq 100\,000$) и на втором — N разделённых пробелами целых чисел: $a_1 a_2 \dots a_N$ ($0 \leq a_i \leq 100\,000$) — состояние игры.

Выходные данные. В единственной строке текстового файла `RK.VAL` вывести длину самого длинного возможного хода.

Пример.	<code>RK.SIS</code>	<code>RK.VAL</code>
	5	2
	1 4 2 3 5	

2. Фокусник

1 секунда 30 очков

В колоде карт фокусника N карт, все карты различны. Для показывания своего любимого фокуса он выбирает из колоды комплект из K карт. Фокусник очень суеверен и считает, что если использовать один комплект карт дважды то трюк не выйдет. Необходимо написать программу, которая определяла бы сколько раз может иллюзионист показывать свой фокус не рискуя провалом.

Входные данные. На единственной строке текстового файла `MK.SIS` два разделённых пробелом числа: размер колоды N ($1 \leq N \leq 200$) и количество карт необходимое для трюка K ($1 \leq K \leq N$).

Выходные данные. На единственной строке текстового файла `MK.VAL` вывести число различных комплектов из K карт, которые фокусник может составить из своей колоды.

Пример.	<code>MK.SIS</code>	<code>MK.VAL</code>
	5 3	10

Обозначим 5 карт в колоде буквами A, B, C, D и E . Возможные комплекты из 3-х карт в таком случае следующие: $\{A, B, C\}, \{A, B, D\}, \{A, B, E\}, \{A, C, D\}, \{A, C, E\}, \{A, D, E\}, \{B, C, D\}, \{B, C, E\}, \{B, D, E\}$ ja $\{C, D, E\}$.

3. Электропитание

1 секунда

50 очков

Компоненты электрической схемы размещены на плате из изолирующего материала. Соединения между компонентами выполнены в виде прямых «дорожек». Если две дорожки имеют общую точку, то в этой точке есть и электрический контакт.

Для того чтобы обеспечить питанием компоненты схемы, необходимо соединить некоторые дорожки с блоком питания. Необходимо написать программу, которая бы находила минимальный набор точек, которые достаточно соединить с блоком питания, для того чтобы передать питание на все заданные дорожки.

Входные данные. На первой строке текстового файла `EL.SIS` дано число дорожек N ($1 \leq N \leq 1000$), которые необходимо обеспечить питанием. На каждой из следующих N строк задано по 4 числа $x_1 y_1 x_2 y_2$: координаты концов каждой из дорожек. Все координаты — целые числа в диапазоне $0 \dots 1000$.

Выходные данные. На первой строке текстового файла `EL.VAL` вывести число точек K , которые необходимо соединить с блоком питания. На каждую из следующих K строк вывести два числа, разделённых пробелом: координаты точки на плате, которую надо соединить с питанием. С питанием можно соединять только конечные точки дорожек.

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

Как видно из рисунка, все дорожки соединены между собой, поэтому для того чтобы обеспечить их всех питанием достаточно соединить с блоком питания одну точку — это может быть любая из шести точек, заданных во входном файле.

