

1. Подпоследовательность

5 секунд

30 очков

Написать программу, которая находит в последовательности чисел возрастающую подпоследовательность максимальной длины.

Входные данные. На первой строке текстового файла `ajada.sis` находится количество элементов в последовательности N ($1 \leq N \leq 1000$) и на второй строке отделённые друг от друга пробелами её элементы (целые числа, не превышающие 10 000 по абсолютному значению).

Выходные данные. На первую строку текстового файла `ajada.val` вывести длину найденной подпоследовательности K и на вторую строку — её элементы. Если подпоследовательностей с максимальной длиной несколько, то вывести самую правую из них.

Пример.	<code>ajada.sis</code>	<code>ajada.val</code>
	7	3
	1 2 4 4 3 4 7	3 4 7

2. Отрезок

1 секунда

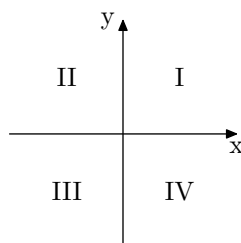
30 очков

На координатной плоскости находится отрезок. Известны его вершины A и B . Написать программу, которая находит, с какими четвертями координатной плоскости у этого отрезка есть общие точки.

Входные данные. На первой строке текстового файла `aloik.sis` находятся координаты точки A — x_A и y_A , а на второй строке координаты точки B — x_B и y_B . Все координаты целые числа, не превышающие 10 000 по абсолютному значению.

Выходные данные. Вывести на первые четыре строки текстового файла `aloik.val` слова JAH (ДА по-эстонски) или EI (НЕТ по-эстонски) в зависимости от того, имеет ли отрезок общие точки с соответственно первой, второй, третьей или четвёртой четвертями.

Примечание. Четверти пронумерованы в очереди против хода часов начиная с правой верхней, как показано на рисунке.



Примечание. Точки лежащие на координатных осях не принадлежат ни одной четверти.

Пример.	<code>aloik.sis</code>	<code>aloik.val</code>
	2 3	JAH
	5 -7	EI
		EI
		JAH

Пример.	<code>aloik.sis</code>	<code>aloik.val</code>
	1 -3	EI
	-2 1	JAH
		JAH
		JAH

3. Путешествие на самолёте

1 секунда

40 очков

Бизнесмен хочет перелететь на самолёте из города A в город C . Так как между этими городами нет прямой линии, приходится в городе B делать пересадку. На пересадку уходит как минимум 30 минут. Поездка может длиться несколько последовательных дней, а самолёт может находиться в воздухе и в полночь. Написать программу, которая на основе графиков перелётов $A - B$ и $B - C$ строит план поездки с минимальными временными затратами.

Входные данные. На первой строке текстового файла `areis.sis` находится N_1 — количество перелётов на линии $A - B$ ($1 \leq N_1 \leq 100$). На следующих N_1 строках для каждого перелёта записано время вылета из A и прибытия в B . На следующей строке находится N_2 — число перелётов на линии $B - C$ ($1 \leq N_2 \leq 100$). На следующих N_2 строках для каждого перелёта записано время вылета из B и прибытия в C . Каждый полёт длится как минимум 1 минуту и как максимум 24 часа.

Выходные данные. Вывести в текстовый файл `areis.val` план полёта с наименьшими затратами времени. На первую строку файла вывести времена вылета и прибытия для перелёта из A в B , а на вторую из B в C . Если решений несколько, вывести любое из них.

Пример.	<code>areis.sis</code>	<code>areis.val</code>
	2	12:20 15:50
	8:10 10:20	17:10 19:20
	12:20 15:50	
	1	
	17:10 19:20	