

1. Точки на прямой

3 секунды

20 очков

На плоскости даны N точек (все точки разные). Написать программу для нахождения максимального множества точек находящихся на одной прямой.

Входные данные. На первой строке текстового файла `sirge.sis` число точек N ($1 \leq N \leq 100$) и на следующих N строках находятся координаты точек: на каждой строке по два целых числа X_i Y_i , разделённых пробелом ($1 \leq i \leq N$, $|X_i| \leq 10\,000$, $|Y_i| \leq 10\,000$). Считаем, что точки пронумерованы $1 \dots N$ в порядке их нахождения во входном файле.

Выходные данные. На первой строке текстового файла `sirge.val` вывести K , максимальное число точек, находящихся на одной прямой, а на второй строке вывести K разделённых пробелами числа: номера этих точек в возрастающем порядке.

Пример.	<code>sirge.sis</code>	<code>sirge.val</code>
	5	3
	3 2	2 4 5
	1 2	
	1 4	
	3 4	
	6 7	

Оценивание. В этом задании 50% очков получают решения, которые смогут найти только максимальное число точек на одной прямой, но не сами точки.

2. Лототрон

1 секунда

40 очков

Лототрон состоит из N кругов, находящихся друг на друге. На каждом из кругов соответственно S_1, S_2, \dots, S_N секторов. Все круги вращаются одновременно в одном направлении и на то чтобы прокрутиться на один сектор вперёд каждому кругу уходит одна единица времени. На каждом круге один сектор отмечен. В момент времени $t = 0$ в отмеченный сектор верхнего круга кладётся шарик и все круги начинают вращаться.

Ещё в лототроне есть специальный вертикальный туннель между кругами. Если в какой-то момент времени отмеченные сектора двух соседних кругов находятся непосредственно друг над другом, там же где этот туннель, то шарик может упасть из отмеченного сектора верхнего круга в отмеченный сектор нижнего круга. Если одновременно друг над другом в месте туннеля оказываются отмеченные сектора более чем двух кругов, шарик может упасть насквозь и более чем на один “этаж” вниз. Можно считать, что падение шарика происходит мгновенно. С нижнего круга лототрона шарик уже никуда не упадёт.

Требуется написать программу, которая сможет определить максимальное количество этажей на которое упадёт шарик и сколько времени на это уйдёт.

Входные данные. На первой строке текстового файла `loto.sis` число кругов N ($1 \leq N \leq 1\,000$). Круги пронумерованы числами $1 \dots N$ сверху вниз. На каждой из следующих N строк два разделённых пробелом целых числа S_i и P_i : число секторов на i -ом круге лототрона и начальное расстояние отмеченного сектора до позиции туннеля, измеренное в секторах в сторону вращения кругов ($1 \leq i \leq N$, $0 \leq P_i < S_i \leq 1\,000$).

Выходные данные. На единственной строке текстового файла `loto.val` вывести два разделённых пробелом числа I и T , где I — номер круга, дальше которого шарик уже не доберётся, и T — момент времени в который шарик попадёт на этот круг.

Пример.	<code>loto.sis</code>	<code>loto.val</code>
	4	3 6
	3 1	
	5 1	
	4 2	
	8 3	

В приведённом примере отмеченный сектор круга 1 доходит до туннеля в момент времени $t = 1$. В

тот же момент у туннеля находится и отмеченный сектор круга 2, поэтому шарик падает на второй круг (но не дальше т.к. отмеченный сектор круга 3 к этому времени не находится у туннеля). Далее второй круг с шариком делает полный оборот и шарик снова подходит в позицию туннеля к моменту времени $t = 6$. В этот момент находится у туннеля отмеченный сектор круга 3, куда шарик и падает. Несложно заметить, что на четвертый круг шарик никогда не попадёт.

3. Двоичный сдвиг

1 секунда

40 очков

Двоичным (битовым) сдвигом данного двоичного числа на один бит влево называют число, которое получается если дописать к изначальному числу справа бит 0.

Операция “исключающее или” (которую также называют XOR и обозначают как \oplus) является логической операцией на двух логических величинах, результат которой “верно”, если ровно один из операндов равен “верно”, и “неверно” во всех остальных случаях.

При “сложении” двоичных чисел операцией \oplus , их биты рассматривают как логические величины (1 = “верно”, 0 = “неверно”) и каждый бит результата считается как XOR соответствующих битов аргументов (младший бит результата — это XOR младших битов аргументов, второй бит результата — XOR вторых битов аргументов, и т.д.). Например, $0101_2 \oplus 1100_2$ считается следующим образом: младший (самый правый) бит результата равен $1 \oplus 0 = 1$; следующий бит равен $0 \oplus 0 = 0$; далее $1 \oplus 1 = 0$; и наконец $0 \oplus 1 = 1$. В итоге получаем $0101_2 \oplus 1100_2 = 1001_2$. Так как \oplus -сумма двух битов всегда один бит, переводов в следующий разряд не бывает.

Пусть дано n -битное число A . Рассмотрим его двоичные сдвиги s_0, s_1, \dots, s_{n-1} , где s_0 означает само число A , и для каждого $i > 0$, s_i — это сдвиг числа s_{i-1} на один бит влево. Например, если $A = 1110_2$, то

$$\begin{aligned}s_0 &= 1110_2, \\s_1 &= 11100_2, \\s_2 &= 111000_2, \\s_3 &= 1110000_2.\end{aligned}$$

Далее рассмотрим \oplus -суммы чисел s_i : $x_0 = s_0$, и для каждого $i > 0$, $x_i = x_{i-1} \oplus s_i$. К примеру, если $A = 1110_2$, то

$$\begin{aligned}x_0 &= 1110_2, \\x_1 &= 1110_2 \oplus 11100_2 = 10010_2, \\x_2 &= 10010_2 \oplus 111000_2 = 101010_2, \\x_3 &= 101010_2 \oplus 1110000_2 = 1011010_2.\end{aligned}$$

Написать программу, которая по заданному $B = x_{n-1}$ восстанавливает изначальное число A .

Входные данные. На единственной строке текстового файла `bitid.sis` находится $(2n-1)$ -битовое двоичное число B ($1 \leq n \leq 100$). B может начинаться с нулей.

Выходные данные. На первой строке текстового файла `bitid.val` вывести слово JAH, если найдётся такое n -битовое число A , из которого можно получить описанным образом B , или EI, если такого числа не найдётся. Если число A найдётся, вывести его на второй строке файла, причём в точности n -битовым (выводимое число может начинаться с нулей).

Пример.	<code>bitid.sis</code>	<code>bitid.val</code>
	1011010	JAH
		1110

Пример.	<code>bitid.sis</code>	<code>bitid.val</code>
	10110	EI

Оценивание. В этом задании за тесты с ответом EI получают очки только те программы, которые пройдут хотя бы половину тестов с ответом JAH.