

1. Встреча

1 секунда

30 очков

На шоссе между городами А и В находится N бензозаправок, и в обоих городах также есть заправки. Заправки пронумерованы числами $0 \dots N + 1$ в направлении от города А к городу В. Два друга начинают одновременно двигаться — один из А в В, а другой из В в А — и движутся навстречу друг другу с одинаковой постоянной скоростью.

Написать программу, которая проверяет, встретятся ли друзья на какой-то заправке или между заправками.

Входные данные. В первой строке текстового файла `koht.sis` дано количество заправок на шоссе, N ($1 \leq N \leq 10\,000$). Во второй строке дано $N + 1$ целых чисел A_i ($0 < A_i \leq 100$): расстояния от заправочной станции i до станции $i + 1$ ($0 \leq i \leq N$).

Выходные данные. В первую строку текстового файла `koht.val` вывести слово JAH, если друзья встретятся на одной из заправок, или EI, если между заправками. Если они встретятся на заправке, вывести во вторую строку файла номер этой заправки.

Пример.	<code>koht.sis</code>	<code>koht.val</code>
	2	JAH
	1 1 2	2

Пример.	<code>koht.sis</code>	<code>koht.val</code>
	3	EI
	1 4 3 5	

Оценивание. В этом задании за тесты с ответом EI очки получают только те программы, которые правильно решат хотя бы один тест с ответом JAH.

2. Палиндромы

1 секунда

30 очков

Палиндромом называется слово, которое одинаково читается слева направо и справа налево. Например, *наган* — это палиндром, а *нарзан* — нет.

Написать программу, которая проверяет, можно ли получить из данных текстов палиндромы, если из них удалить все символы, кроме букв (цифры, пробелы, дефисы и т.д.).

Входные данные. В первой строке текстового файла `pal.sis` дано количество текстов, N ($1 \leq N \leq 100$), и в каждой из следующих N строк дан текст, состоящий из латинских букв, цифр, пробелов и дефисов. Длина каждого текста не превышает 100 знаков.

Выходные данные. В текстовый файл `pal.val` вывести ровно N строк. В строку номер i ($1 \leq i \leq N$) вывести слово JAH (“да”) или EI (“нет”) в зависимости от того, превращается ли текст в строке номер $i + 1$ входного файла в палиндром. Регистр букв не должен учитываться (большие и маленькие буквы считать одинаковыми).

Пример.	<code>pal.sis</code>	<code>pal.val</code>
	2	JAH
	kirik	EI
	kirjak	

3. Последовательность

1 секунда

30 очков

Рассмотрим последовательность чисел, в которой каждый элемент “описывает” предыдущий. Последовательность строится следующим образом: $A_1 = 1 \sim$ “одна единица” $\rightarrow A_2 = 11 \sim$ “две единицы” $\rightarrow A_3 = 21 \sim$ “одна двойка, одна единица” $\rightarrow A_4 = 1211 \sim$ “одна единица, одна двойка, две единицы” $\rightarrow A_5 = 111221 \sim$ “три единицы, две двойки, одна единица” $\rightarrow A_6 = 312211 \dots$

Выбирая в качестве первого элемента A_1 разные числа, мы можем получать различные последовательности.

Написать программу, которая проверяет, встречается ли данное число B в какой-нибудь последовательности A , составленной по описанным правилам из однозначного числа A_1 .

Входные данные. В единственной строке текстового файла `jada.sis` дано число B ($0 \leq B < 2^{63}$).

Выходные данные. В первую строку текстового файла `jada.val` вывести слово ЯН (“да”) или ЕИ (“нет”) в зависимости от того, встречается ли число B в какой-нибудь последовательности A . Если да, то во вторую строку вывести два разделённых пробелом целых числа: первый член последовательности, A_1 ($0 \leq A_1 \leq 9$), и индекс числа B в этой последовательности.

Пример.	<code>jada.sis</code>	<code>jada.val</code>
	11	ЯН
		1 2

Взяв $A_1 = 1$, получаем $A_2 = 11 = B$.

Пример.	<code>jada.sis</code>	<code>jada.val</code>
	132116	ЯН
		6 5

Взяв $A_1 = 6$, получаем $A_2 = 16$, $A_3 = 1116$, $A_4 = 3116$, $A_5 = 132116 = B$.

Пример.	<code>jada.sis</code>	<code>jada.val</code>
	444	ЕИ

Оценивание. В этом задании за тесты с ответом ЕИ очки получают только те программы, которые правильно решат хотя бы один тест с ответом ЯН.

4. Морской бой

1 секунда

30 очков

“Морской бой” — это настольная игра, в которой играют на игровом поле размером 10×10 клеток. В начале игры участник располагает на поле свой “флот”, состоящий из 4-палубного, двух 3-палубных, трёх 2-палубных и четырёх 1-палубных кораблей. Каждый корабль обозначает горизонтальная или вертикальная полоса, состоящая из $n \times 1$ квадратов, где n — количество палуб корабля. Все корабли должны располагаться целиком на игровом поле и не должны пересекаться или касаться ни краями, ни углами.

Написать программу, которая считывает описание расположения девяти кораблей и добавляет туда десятый корабль, так чтобы полученное расположение отвечало всем требованиям игры.

Входные данные. В текстовом файле `laev.sis` ровно 10 строк, в каждой строке ровно 10 знаков, среди которых точка (.) обозначает пустую клетку и решётка (#) обозначает клетку, занятую каким-то кораблём.

Выходные данные. В текстовый файл `laev.val` вывести ровно 10 строк, в каждой строке ровно по 10 знаков: дополненное описание игрового поля в том же формате. Существующие корабли двигать нельзя, новый корабль необходимо расположить на свободных клетках. Можно предполагать, что во всех тестах расположение нового корабля возможно. Если возможностей расположения нового корабля несколько, вывести любую из них.

Пример.	<code>laev.sis</code>	<code>laev.val</code>
	<code>...#.....</code>	<code>...#.....</code>
	<code>.....</code>	<code>.....</code>
	<code>..###.#.#.</code>	<code>..###.#.#.</code>
	<code>.....#...</code>	<code>.....#...</code>
	<code>.###.....</code>	<code>.###.....</code>
	<code>.....#..</code>	<code>.....#..</code>
	<code>.#.....</code>	<code>.#.#.....</code>
	<code>.#....#..</code>	<code>.#...#..</code>
	<code>.#.....</code>	<code>.#.....</code>
	<code>....##.....</code>	<code>....##.....</code>

5. Радиопередача

1 секунда

30 очков

На одной радиостанции есть передача, в которой играют песни — новинки недели. В передаче нет рекламных пауз или диктора, поэтому эфир передачи необходимо точно заполнить от начала до конца. Поскольку речь идёт только о новых песнях, ни одну песню нельзя играть более одного раза.

Написать программу, которая определяет, возможно ли из данных песен составить передачу требуемой длительности.

Входные данные. В первой строке текстового файла `raadio.sis` дано количество новых песен, N ($0 \leq N \leq 1000$). В каждой из следующих N строк дана длительность одной песни в формате $M:S$, где M — это количество минут, и S — количество секунд (двухзначное число). Песни пронумерованы числами $1 \dots N$ в порядке их следования в файле. Длина каждой песни не превышает 6 минут. В последней строке файла дана длительность передачи, которая не превышает 3 часа (180 минут), также в формате $M:S$.

Выходные данные. В первую строку текстового файла `raadio.val` вывести слово JAH (“да”) или EI (“нет”) в зависимости от того, возможно ли составление передачи требуемой длины из данных песен. В случае положительного ответа во вторую строку файла вывести K , число песен, играющих в передаче, и в третью строку вывести K разделённых пробелами целых чисел: номера песен в возрастающем порядке. Если подходящих комплектов песен несколько, вывести любой из них.

Пример.	<code>raadio.sis</code>	<code>raadio.val</code>
	<code>4</code>	<code>JAH</code>
	<code>3:28</code>	<code>3</code>
	<code>2:42</code>	<code>1 3 4</code>
	<code>3:16</code>	
	<code>4:02</code>	
	<code>10:46</code>	

Пример.	<code>raadio.sis</code>	<code>raadio.val</code>
	<code>4</code>	<code>EI</code>
	<code>3:20</code>	
	<code>3:12</code>	
	<code>3:34</code>	
	<code>2:56</code>	
	<code>8:45</code>	

Оценивание. В этом задании за тесты с ответом EI очки получают только те программы, которые правильно решат хотя бы один тест с ответом JAH.

6. Симметрия

30 очков

Рассмотрим изображение, состоящее из $N \times M$ чёрных и белых пикселей.

Осью симметрии изображения назовём прямую, для которой у каждого чёрного пикселя найдётся другой чёрный пиксель, такой что ось симметрии является серединным перпендикуляром отрезку, соединяющему эти два пикселя. В качестве исключения считаем все пиксели, лежащие на оси симметрии, симметричными самим себе.

Центром симметрии изображения назовём точку, для которой у каждого чёрного пикселя найдётся другой чёрный пиксель, такой что центр симметрии является центром отрезка, соединяющего эти два пикселя. В качестве исключения считаем пиксель, находящийся в центре симметрии, симметричным самому себе.

В обоих случаях белые точки могут оставаться без пары.

Найти все оси симметрии и центр симметрии данного изображения (либо выяснить, что их нет).

Входные данные. В первой строке текстового файла `kuju.sis` дано количество строк N ($1 \leq N \leq 1000$) и столбцов M ($1 \leq M \leq 1000$) изображения. В каждой из следующих N строк дано ровно M знаков: описание изображения, где точка (.) обозначает белый, а решётка (#) — чёрный пиксель. Строки изображения пронумерованы $1 \dots N$ сверху вниз, а столбцы пронумерованы $1 \dots M$ слева направо.

Выходные данные. В текстовый файл `kuju.val` вывести ровно пять строк. В четыре первые строки вывести информацию об осях симметрии: горизонтальной, вертикальной, наклонной под углом 45° снизу вверх (из левого нижнего угла в направлении правого верхнего угла) и наклонной под углом 45° сверху вниз (из верхнего левого угла в направлении правого нижнего). В пятую строку вывести информацию о центре симметрии.

В каждую строку вывести слово ЯАН (“да”) или ЕИ (“нет”) в зависимости от того, есть ли у данного изображения соответствующая ось или центр симметрии. За словом ЯАН вывести координаты одной точки оси симметрии или центра симметрии (сначала номер строки, затем номер столбца).

Пример.	<code>kuju.sis</code>	<code>kuju.val</code>
	3 4	ЯАН 2 2
	..#.	ЯАН 1 3
	.###	ЯАН 3 2
	..#.	ЯАН 1 2
		ЯАН 2 3

Пример.	<code>kuju.sis</code>	<code>kuju.val</code>
	4 4	ЕИ
	..#.	ЯАН 1 2.5
	..#.	ЕИ
	####	ЕИ
	..#.	ЕИ

Оценивание. В этом задании даны 10 комплектов входных данных в файлах `kjutest.01.sis` ... `kjutest.10.sis`. В качестве решения предоставить соответствующие входные данные в файлах `kjutest.01.val` ... `kjutest.10.val`. Текст программы предоставлять не требуется, он не оценивается.