

**1. ТАБЛИЦА БАЛЛОВ**

10 очков

5 секунд

Как уже известно, в этом году на олимпиаде используется новая система оценивания: для решения предоставляется 5 задач, но в зачет каждому участнику идут только результаты 3 лучших решений. Для использования новой системы нужна новая программа составления протоколов. Написать программу, которая получает результаты всех 5 программ каждого участника и составляет таблицу, где окончательный результат каждого участника – это сумма баллов 3 лучших заданий.

Входные данные. На первой строке текстового файла PUNKTID.SIS число участников  $N$  ( $N \leq 100$ ) и на следующих  $2 \times N$  строках данные о каждом участнике на 2 строках: на первой строке имя участника (не более 30 символов) и на второй строке баллы заработанные за каждое задание (5 разделенных пробелами неотрицательных чисел, значение которых не превышает 100).

Выходные данные. В текстовый файл PUNKTID.VAL вывести ровно  $N$  строк, на каждую строку имя участника и окончательное количество баллов, разделенные пробелом. Строки выводить в порядке убывания баллов. Участников с одинаковым количеством баллов выводить в том же порядке, как они были в входном файле.

Пример.

PUNKTID.SIS	PUNKTID.VAL
3	Karu, Mati 41
Karu, Kati	Karu, Kati 35
10 10 15 0 2	Rebane, Rein 34
Rebane, Rein	
10 10 12 12 10	
Karu, Mati	
0 0 14 15 12	

**2. СКОБОЧНОЕ ВЫРАЖЕНИЕ**

20 очков

5 секунд

При проверке синтаксиса выражения, кроме всего остального, необходимо проверить правильность расположения скобок. Для этого мы можем исследовать строку, которая образуется при удалении всех остальных символов из выражения. Скобочное выражение - это выражение состоящее только из скобок. Правильное скобочное выражение определяется следующим образом:

- пустая строка – это корректное скобочное выражение;
- если  $A$  корректное скобочное выражение, то  $(A)$ ,  $[A]$  и  $\{A\}$  тоже корректные скобочные выражения;
- если  $A$  и  $B$  корректные скобочные выражения, тогда  $AB$  также корректное скобочное выражение;
- никакие другие выражения не являются корректными скобочными выражениями.

Написать программу, которая определяет для каждого данного ей скобочного выражения, корректное ли оно.

Входные данные. На первой строке текстового файла SULUD.SIS находится количество выражений  $N$  ( $N \leq 20$ ) и на каждой из следующих  $N$  строк одно выражение длиной не более 80 символов.

Выходные данные. В текстовый файл SULUD.VAL вывести ровно  $N$  строк – одну на каждое слово из входного файла. На строчку под номером  $i$  вывести JAH, если на  $i+1$  строке входного файла находится корректное скобочное выражение и вывести EI в противном случае.

Пример.

SULUD.SIS	SULUD.VAL
3	EI
()	JAH
{ () [] }	EI
{ {} }	

### 3. ТРУБЫ

20 очков

5 секунд

На заводе по изготовлению жестяных труб находится некоторое количество труб с четырёхугольными поперечными сечениями. Рабочий хочет поместить в хранилище трубы, поперечное сечение которых имеет форму квадрата. Так как в хранилище места мало, он также хочет вставить как можно больше труб друг в друга. Так как все трубы одинаковой длины, рабочий должен при этом учитывать только поперечное сечение. У рабочего для каждой трубы есть чертеж, на котором видны координаты вершин поперечного сечения. Написать программу, которая найдет среди всех труб те, которые имеют квадратное сечение и образуют самую длинную последовательность в которой каждая последующая труба помещается в предыдущую. Труба помещается в другую если длина ее сторон меньше длины сторон другой трубы.

Входные данные. На первой строке текстового файла TORUD.SIS число труб  $N$  ( $N \leq 100$ ) и на каждой из следующих  $N$  строк координаты перпендикулярного сечения одной из труб (8 разделенных пробелами целых чисел, абсолютное значение которых не превышает 1000). Координаты вершин даны по часовой стрелке. Трубы пронумерованы от 1 до  $N$  в том же порядке как они представлены в входном файле.

Выходные данные. На первую строку файла PUNKTID.VAL вывести количество труб с квадратным поперечным сечением. На вторую строку файла вывести количество труб поместившихся друг в друга. На третью строку вывести последовательность номеров этих труб, от большей к меньшей. Если последовательностей с максимальной длиной несколько, вывести любую из них.

Пример.

TORUD.SIS	TORUD.VAL
4	2
5 7 5 10 12 10 12 7	2
5 6 5 3 2 3 2 6	4 2
10 0 4 0 7 3 13 3	
12 7 8 4 5 8 9 11	

Оценивание. В этом задании при оценивании отдельно учитывается решение первой части задания. Если Ваша программа умеет считать только количество труб с квадратным поперечным сечением, тогда выведите на вторую строку выходного файла -1 и оставьте третью строку пустой. За такое решение можно получить до 10 очков.

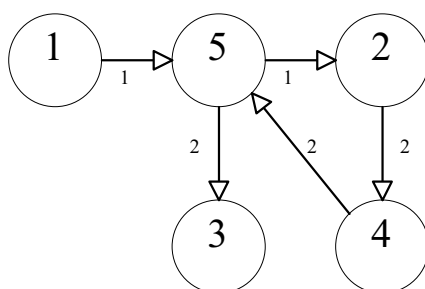
#### 4. ПЕЩЕРЫ МИНОТАВРА

40 очков

10 секунд

Когда в Древней Греции Тезей пошел в лабиринт искать Минотавра, Ариадна дала ему клубок ниток, который Тезей в лабиринте раскручивал и с помощью которого потом мог найти обратный путь. Лабиринт состоит из пещер соединенных между собой проходами. Каждый проход соединяет ровно две пещеры и между любыми двумя пещерами может быть не больше одного прохода.

Тезей, ищущий Минотавра, никогда не заходит в проход где уже есть нитка; но он может заходить в пещеру где есть нитка если через нее проходит какой нибудь проход без нитки. Если Тезей дойдет до тупика (пещера, где больше нет непросмотренных проходов), то он сварачивая клубок идет обратно и ищет новые проходы. Написать программу, которая находит какой длины должен быть клубок, чтобы Тезею его хватило на любой разрешенный путь.



Входные данные. На первой строке текстового файла KOOPAD.SIS находится число пещер N (N≤100) и число проходов M (M≤1000). На каждой из следующих M строк файла для каждого прохода находятся номера пещер, которые соединены этим проходом и длина этого прохода. Все пещеры пронумерованы от 1 до N и номером 1 обозначена пещера в которой находится единственный вход в лабиринт. Длина прохода – целое число не превышающее 30.

Выходные данные. На первую строку файла KOOPAD.VAL вывести искомую длину клубка и на второй ряд разделенные пробелами номера пещер, через которые проходит путь, при котором клубок разматывается до конца. Номера пещер выводить начиная со входа в лабиринт.

Пример.

KOOPAD.SIS	KOOPAD.VAL
5 5	8
2 4 2	1 5 2 4 5 3
1 5 1	
2 5 1	
5 4 2	
3 5 2	

Оценивание. При тестировании этого задания корректное решение (выводимый путь не проходит не одного прохода больше одного раза) получит баллов пропорционально квадрату соотношению найденной решением длины пути и длины пути найденного решением жюри. Например если Ваше решение найдет путь 5 метров а решение жюри 10 метров, тогда Ваше решение получит  $(5/10)^2 = 0,25 = 25\%$  от количества баллов этого теста. Максимально можно получить 150% от количества баллов за тест. Все суммы баллов округляются к большему целому числу.

### 5. ПРЕФИКС-СУФИКС

40 очков

2 секунды

Строку  $S_1$  длиной  $P_1$  называют префиксом строки  $S_2$ , если длина  $S_1$  не превышает длину  $S_2$  и первые  $P_1$  символов  $S_2$  совпадают с соответствующими символами  $S_1$ . Аналогично строку  $S_1$  длиной  $P_1$  называют суффиксом строки  $S_2$ , если длина  $S_1$  не превышает длину  $S_2$  и последние  $P_1$  символов  $S_2$  совпадают с соответствующими символами  $S_1$ . Очевидно, что каждое слово является самому себе как префиксом, так и суффиксом. Префикс строки называют собственным префиксом, если он короче самой строки. Аналогично суффикс строки называют собственным суффиксом, если он короче самой строки.

Написать программу, которая находит для каждого префикса  $S_1$  данной строки  $S$  собственный суффикс  $S_2$  максимальной длины, который одновременно является префиксом строки  $S$ . (О том, где в практике используется данная конструкция, вы узнаете после соревнования во время объяснения решений.)

Входные данные. На первой строке текстового файла `PREFIKS.SIS` находится длина  $P$  ( $P \leq 20000$ ) строчки  $S$ . На следующих  $P$  строках файла находится сама строчка, по одному символу на каждой строке.

Выходные данные. В текстовый файл `PREFIKS.VAL` вывести ровно  $P$  цтрок. На строку под номером  $i$  вывести длину искомого суффикса  $i$ -буквенного префикса строчки  $S$ .

Пример.

<code>PREFIKS.SIS</code>	<code>PREFIKS.VAL</code>
б	0
а	1
а	0
b	1
а	2
а	0
с	