1. **ПРЯМОУГОЛЬНИКИ** 5 очков 10 секунд

Написать программу, которая находит на основе координат одной точки и вершин некоторых прямоугольников количество прямоугольников, которым она принадлежит. Стороны прямоугольников параллельны координатным осям.

Входные данные. На первой строке файла `RISTK.SIS` записаны координаты точки `Xp Yp`. На второй строке – количество прямоугольников `N` (`1≤N≤100`) и на следующих `N` строках записаны координаты левых нижних и правых верних вершин прямоугольников X_{i1} Y_{i1} X_{i2} Y_{i2}. Все координаты – целые числа, абсолютная величина которых не превышает `10000`. Числа отделены друг от друга пробелами.

Выходные данные. На единтсвенную строку файла `RISTK.VAL` вывести количество прямоугольников, которые содержат указанную точку. Точка, лежащая на стороне или вершине прямоугольника считается принадлежашей ему.

Пример.

```
RISTK.SIS          RISTK.VAL
1 1                2
3
-1 0 4 3
1 1 6 7
-2 2 7 4
```

2. **ПОСЛЕДОВАТЕЛЬНОСТИ БИТОВ** 10 очков 10 секунд

Написать программу, которая находит все последовательности битов, которые содержат заданное количество нолей и единиц.

Входные данные. На единственной строке файла `JADAD.SIS` записаны разделённые пробелом целые числа `M` и `N` (`1≤M+N≤15`). `M` – количество нолей, `N` – количество единиц.

Выходные данные. Вывести в файл `JADAD.VAL` все такие последовательности битов, в которых ровно `M` нолей и `N` единиц. Каждая последовательность должна быть выведена на отдельной строке, причём биты должны следовать один за другим. Порядок последовательностей не важен, но каждая последовательность должна быть выведена ровно один раз.

Пример.

```
JADAD.SIS          JADAD.VAL
2 1                100
                   010
                   001
```

3. **SXML** 20 очков 10 секунд

Язык SXML (*Simplified XML* – англ.) – это упрощённая версия языка разметки текста XML. SXML-последовательность состоит из слов, между которыми располагаются разделители и метки. SXML-слова состоят из больших и маленьких латинских букв и имеют положительную длину. SXML-разделителями являются пробел, табулятор (код ASCII 9) и перевод строки. SXML-фраза определяется следующими правилами:

- пустая строка (пустой стринг) является SXML-фразой;
- если `F` – SXML-фраза, а `G` – SXML-слово или SXML-разделитель, то `FG` является SXML-фразой;
- если `F` – SXML-фраза, а `G` – SXML-слово, то `<G>F</G>` является SXML-фразой; `<G>` называют меткой начала, `</G>` меткой конца, а слово `G` названием меток.

СХМЛ-последовательность определяется следующим правилом:

- если F является SXML-фразой, а G1 и G2 – (возможно пустые) последовательности SXML-разделителей, то G1<SXML>F</SXML>G2 является SXML-последовательностью.

Написать программу, которая проверяет, яиляется ли содержимое входного файла SXML-последовательностью.

Входные данные. В файле SXML.SIS находится до 100000 символов, причём в каждой строке не более 100 символов. Известно, что название ни одой метки не состоит из более чем 50 символов, а глубина вложенности меток не превышает 100.

Выходные данные. Если файл является корректной SXML-последовательностью, то вывести в файл SXML.VAL на единственнкую строку слово JAH. Если же файл не является корректной SXML-последовательностью, то вывести на первую строку файла SXML.VAL слово EI, а на вторую строку два разделённых пробелом числа: номер строки и позиции такога символа, что всё содержимое файла до него является началом некоторой SXML-последовательности, а включая его – нет. Отсчёт обеих чисел начинается с одного. Если во входном файле присутствует перевод строки в неразрешённом месте, то вывести номер этой строки и ноль. Если файл кончается слишком рано (т.е. конец файла в неразрешённом месте), то вывести два ноля.

Пример 1.

```
SXML.SIS                                SXML.VAL
<SXML>                                  JAH
   See on SXML test
<aaa>testtesttest</aaa>
</SXML>
```

Пример 2.

```
SXML.SIS                                SXML.VAL
  <SXML>AAA<AAA>xxx                     EI
yyy zzz<bb>misiganes                    3 12
</bb>   </AaA></SXML>
```

Пример 3.

```
SXML.SIS            SXML.VAL
<SXML>Poolik fail</SX                   EI
                                        0 0
```

4. **ВЫБОР ФАЙЛА** 30 очков 10 секунд

В диалоге открытия файла выводятся в алфавитном порядке названия всех находящихся в активном каталоге файлов. Для выбора файла используются клавиши-стрелки и клавиши-буквы.

В начале диалога выбранным является первый в списке файл. При нажатии клавиш-букв из списка выбирается первый файл, начало названия которого совпадает с введённой последовательностью. Если такой отсутствует, то соответствующее нажатие клавиши игнорируется. При нажатии клавиши <UP> выбирается файл предшествующий в списке выбранному. Если был выбран первый файл, то выбор перемещается на последний. При нажатии клавиши <DOWN> выбирается файл следующий в списке за выбранным. Если был выбран последний файл, то выбор перемещается на первый. При нажатии клавиши-стрелки диалог забывает предыдущие нажатия клавиш-букв и если потом снова нажимают клавишу-букву, то выбор начинется сначала.

Например, если в каталоге находятся файлы aaa, aba, abb, abc, cab и cbb, то при нажатии клавиш <a> и <b> выбранным становится файл aba. Если после этого нажать клавишу <DOWN>, то выбанным станет файл abb и если после этого нажать <c>, то выбранным станет файл cab (а не abc).

Написать программу, которая находит минимальную последовательность нажатий клавиш, после которой выбранным окажется заданный файл.

Входные данные. На первой строке файла FAILID.SIS записано количество файлов N (1≤N≤1000). На следующих N строках расположены имена файлов в алфавитоном порядке. Имена файлов состоят из не более чем 20 маленьких латинских букв. На последней строке записано имя искомого файла. В списке это имя всегда присутствует.

Выходные данные. Вывести в текстовой файл FAILID.VAL искомые нажатия клавиш, каждое нажатие – на отдельной строке. Если возможных последовательностей много, то вывести любую.

Пример.

```
FAILID.SIS          FAILID.VAL
8                   <a>
aaaa                <b>
aaab                <UP>
aaac
aaad
aaae
abcd
abdd
bcde
aaae
```

5. **АВТОБУСНЫЕ ЛИНИИ** 35 очков 10 секунд

В городе расположено N автобусных остановок, которые объедены в кольцевые линии так, что каждая остановка присутствует в точности на одной линии. Автобус едет от одной остановки до следующей ровно 1 минуту. В один момент (когда все автобусы стоят на остановках) диспетчер запоминает места расположения автобусов. Написать программу, которая находит количество минут, по прошествии которых все автобусы будут снова на исходных позициях.

Входные данные. На первой строке файла BUSSID.SIS расположено количество остановок N (1≤N≤1000) и на следующих N строках по два разделённых пробелом названия остановок. Строка вида aaa bbb означает, что на какой-то линии остановка bbb следует за остановкой aaa. Названия остановок состоят из маленьких латинских букв и цифр. Их длина не превышает двадцати. Каждая остановка присутствует в точности на одной строке на первой позиции н в точности на одной строке на второй позиции.

Выходные данные. На единственную строку файла BUSSID.VAL вывести наименьшее положительное количество минут, после которых все автобусы вернутся на свои исходные позиции. Это число никогда не будет превышать 2147483647.

Пример.

```
BUSSID.SIS          BUSSID.VAL
5                   6
aaa eee
ddd bbb
ccc aaa
bbb ddd
eee ccc
```