

1. Имена детей в Паунвере

10 очков

10 секунд

Паунвереский ЗАКС желает собрать статистику о том, как родители называют своих детей. За отсутствием лучших идеи затрудики ЗАКСа просто записывали все имена в текстовый файл. Теперь надо на основе этого файла найти, какие имена были даны, а также сколько раз каждое имя использовалось.

Входные данные. На первой строке текстового файла `LAPSED.SIS` – общее число детей N ($1 \leq N \leq 100$), а на каждой из следующих N строк одно имя, состоящее из 1 то 20 заглавных латинских букв.

Выходные данные. В текстовый файл `LAPSED.VAL` вывести в точности одну строку на каждое имя. На каждую строку вывести имя и число его появления во входном файле, разделённые друг от друга пробелом. Порядок строк в файле не имеет значения.

Пример.

| | |
|-------------------------|-------------------------|
| <code>LAPSED.SIS</code> | <code>LAPSED.VAL</code> |
| 7 | ARNO 2 |
| ARNO | JOOSEP 1 |
| JOOSEP | TEELE 2 |
| TEELE | JORH 1 |
| JORH | AIN 1 |
| TEELE | |
| ARNO | |
| AIN | |

2. ЯЗЫК ПРОГРАММИРОВАНИЯ DL

15 очков

10 секунд

DL – это простой язык программирования, в котором встречаются переменные с однобуквенными идентификаторами, целые числа и операция присваивания. Каждую команду в программе DL напишут на отдельную строку в виде

`переменная=число`

или

`переменная=переменная`

Команды всегда выполняют подряд. На правой стороне присваивания может быть только переменная, которой было уже присвоено значение. Нарушение этого правила является ошибкой и выполнение программы немедленно прекращается.

Входные данные. На каждой строке текстового файла `PROG.SIS` находится одна команда языка DL. Имена переменных – заглавные латинские буквы. Константы – целые числа в промежутке $0..1000$. Пробелов во входном файле нет. В файле может быть от 1 до 100 000 строк.

Выходные данные. Если во время выполнения программы произойдёт ошибка, то вывести в текстовый файл `PROG.VAL` одну строку вида

`VIGA REAL x`

где x номер строки, на которой произошла первая ошибка. Если программа закончит работу без ошибок, то вывести в файл `PROG.VAL` одну строку для каждой переменной, которой было присвоено значение. Строки должны иметь вид

`переменная=число`

где число – значение переменной в конце работы программы. Порядок строк в файле не имеет значения.

Пример.

| | |
|-----------------------|-----------------------|
| <code>PROG.SIS</code> | <code>PROG.VAL</code> |
| A=1 | A=1 |
| D=2 | C=2 |
| C=D | D=1 |
| D=A | |

Пример. PROG.SIS PROG.VAL
 A=1 VIGA REAL 2
 B=C
 C=A

3. ИМЕНА ФАЙЛОВ

20 очков

10 секунд

В многих операционных системах команды могут выполняться по отношению к нескольким файлам одновременно, если указать имена файлов при помощи т.н. универсальных символов. Обычно в качестве универсальных символов используют ? и *. При этом символ ? обозначает один, а * любое количество (в том числе и 0) произвольных символов. Команды, которые могут применяться к нескольким файлам одновременно, должны быть способны определить, какие имена файлов соответствуют данному шаблону (шаблон похож на имя файла, но может содержать и универсальные символы).

Входные данные. На первой строке текстового файла FAILID.SIS находится шаблон – последовательность латинских букв и универсальных символов длиной от 1 до 100 символов. На следующей строке число файлов n ($1 \leq n \leq 100$), а на каждой из следующих n строк одно имя файла, состоящая из 1 до 100 латинских букв.

Выходные данные. В текстовый файл FAILID.VAL вывести те имена, которые соответствуют шаблону. При сравнении строчные и прописные буквы считать разными. Имена вывести в порядке их появления во входном файле.

Пример. FAILID.SIS FAILID.VAL
 d*a??t datatxt
 6 dadot
 datatxt datadatadatadat
 datadoc
 dadot
 datadt
 datatxtx
 datadatadatadat

4. НАБЛЮДЕНИЕ СВЁЗД

25 очков

10 секунд

На планете Эйдос время измеряют в *кronосах* и *календосах*, причём *календос* состоит из k *кronосов*. На Эйдосе живёт астроном Зоро Астро, чей рабочий график повторяется в каждом *календосе*. Как и у всех смертных, у Зоро бывают причины, по которым он не может выйти на работу – семейные события, болезни, отпуска и.т.д. К тому же у Зоро есть начальник, который иногда интересуется, когда Зоро появится на работе.

Входные данные. На первой строке текстового файла ZORO.SIS находятся разделённые друг от друга пробелами длина *календоса* в *кronосах* k ($1 \leq k \leq 1000$), число промежутков в течение одного *календоса*, когда Зоро работает m ($1 \leq m \leq 100$), число отсутствий Зоро n ($0 \leq n \leq 1000$) и текущий момент t ($0 \leq t \leq 10^9$). На каждой из следующих m строк записаны времена начала и конца одного промежутка, когда Зоро работает A и B ($0 \leq A < B \leq k$). Времена даны относительно начала *календоса*, отрезки даны в возрастающем порядке и не пересекаются между собой. На каждой из следующих n строк располагаются времена начала и конца одного отсутствия Зоро C и D ($0 \leq C < D \leq 10^9$). Эти времена даны в *кronосах* по абсолютной шкале, промежутки даны в порядке неубывания начал и могут пересекаться. Все числа – целые. Все промежутки времени – полузамкнутые интервалы, т.е. в промежутке $[T_1, T_2)$ входят все такие моменты t , что $T_1 \leq t < T_2$.

Выходные данные. В текстовый файл ZORO.VAL вывести одно целое число – момент времени, когда Зоро появится на работе, то есть первый момент t начиная с момента T ($t \geq T$), когда у Зоро есть рабочее время и не зарегистрировано ни одного отсутствия.

Пример.

| ZORO.SIS | ZORO.VAL |
|-------------|----------|
| 10 1 1 2000 | 2001 |
| 1 9 | |
| 500 639 | |

5. ТРЕУГОЛЬНИКИ

30 очков

10 секунд

На плоскости даны n треугольников и m точек. Найти, какие точки находятся внутри каких треугольников.

Входные данные. На первой строке текстового файла NURGAD.SIS – число треугольников n ($1 \leq n \leq 10$) и число точек m ($1 \leq m \leq 10$). На каждой из следующих n строк координаты вершин одного треугольника в виде

$$A_x \ A_y \ B_x \ B_y \ C_x \ C_y$$

На каждой из следующих m строк координаты одной точки в виде

$$P_x \ P_y$$

Пронумеруем все треугольники номерами 1 до n и все точки номерами 1 до m в порядке их появления во входном файле. Все координаты – целые числа, абсолютная величина которых не превосходит 1000. Площадь всех треугольников отлична от нуля.

Выходные данные. В текстовый файл NURGAD.VAL вывести n строк – одну строку на каждый треугольник. На i -тую строку вывести номера всех точек, которые находятся внутри i -того треугольника. При этом точки, находящиеся на границе или вершине треугольника считать входящими внутрь. Номера на каждой строке вывести в возрастающем порядке.

Пример.

| NURGAD.SIS | NURGAD.VAL |
|----------------|------------|
| 3 3 | 1 3 |
| 0 -1 2 1 0 2 | 3 |
| 1 1 3 2 3 0 | 1 2 |
| -3 1 3 -2 -2 2 | |
| 0 0 | |
| -2 1 | |
| 2 1 | |