

1. СТРОКА

10 очков

10 секунд

Написать программу, которая считает, сколько в данной строке

- a) знаков пунктуации;
- b) слов;
- c) палиндромов.

Словом называем непустую последовательность букв, ограниченную с обеих сторон либо небуквенными символами, либо началом/концом строки. Палиндромом называем слово, которое читается одинаково слева направо и справа налево. Например, слово SADAS является палиндромом, а SALAT нет.

Входные данные. В текстовом файле RIDA.SIS строка длиной до 250 символов, состоящая из строчных и прописных латинских букв, пробелов и знаков (. , : ; ? !). При сравнении строчных и прописных букв не различать.

Выходные данные. В текстовый файл RIDA.VAL вывести три целых числа: число знаков пунктуации, число слов и число палиндромов во входном файле.

Пример.

<code>RIDA.SIS</code>	<code>RIDA.VAL</code>
<code>Aias sadas saia. Saia sadas aias...</code>	<code>4 6 2</code>

Оценивание. В данной задаче каждая подзадача оценивается отдельно. Если Ваша программа какую-либо подзадачу не решает (например, не считает число палиндромов), то выведите в соответствующую позицию число -1.

2. УГОЛ

10 очков

10 секунд

Написать программу, которая вычисляет величину угла (в градусах), определённую двумя данными на плоскости точками (A и B) и началом координатной системы O.

Входные данные. На первой строке текстового файла NURK.SIS два вещественных числа: x- и y-координаты точки A. На второй строке в том же формате даны координаты точки B. Все координаты вещественные числа, абсолютная величина которых не превышает 1000. Ни A, ни B не совпадают с началом координат O.

Выходные данные. В текстовый файл NURK.VAL вывести вещественное число R: угол, на который надо поворачивать луч OA против часовой стрелки, чтоб он совпадал с лучом OB. Величину угла вывести в градусах с точностью не менее одного числа после запятой и она должна соответствовать условию $0 \leq R < 360$.

Пример.

<code>NURK.SIS</code>	<code>NURK.VAL</code>
<code>0.0 2.0</code>	<code>270.0</code>
<code>3.0 0.0</code>	

3. СООБЩЕНИЕ

20 очков

10 секунд

Члены одного секретного общества передают информацию с помощью пейджером. В интересах конспирации каждый из них знает номера пейджером лишь нескольких сообщников. Каждый из них передаёт полученное сообщение всем, номера пейджером которых он знает. Написать программу, которая определяет число людей, которые не получают информацию, которой сначала владеет только один член общества.

Входные данные. На первой строке текстового файла TEADE.SIS число членом общества: N ($1 \leq N \leq 100$). Все члены пронумерованы от 1 до N в порядке вступления в

общество. На следующих n строках файла данные о знаниях членов. На строке $L+1$ информация о знаниях члена номер L в формате

$$K \ L_1 \ L_2 \ \dots \ L_K$$

где K ($0 \leq K < N$) число сообщников, номера пейджеров которых L знает и $L_1 \dots L_K$ номера этих сообщников. На последней строке файла номер инициатора сообщения: x ($1 \leq x \leq N$).

Выходные данные. В текстовый файл `TEADE.VAL` вывести одно целое число: число сообщников, до которых не дойдёт исходящее от x сообщение.

Пример.

<code>TEADE.SIS</code>	<code>TEADE.VAL</code>
4	1
0	
2 1 4	
1 2	
1 1	
2	

4. ФАЙЛЫ

30 очков

10 секунд

Жёсткий диск компьютера состоит из блоков фиксированной длины. На диске с n блоками они пронумерованы от 0 до $n-1$. Чтоб вести учёт за местом на диске, пользуются специальной таблицей (*file allocation table*, FAT), в которой каждому блоку диска соответствует одно число:

- 3 означает дефектный блок (в этом месте диск испорчен);
 - 2 означает свободный (неиспользованный) блок;
 - 1 означает блок, в котором какой-то файл кончается;
 - 0 .. $n-1$ означает блок, принадлежащий какому-то файлу;
- значение числа: номер блока в котором этот файл продолжается.

Кроме того имеется ещё одна таблица, в которой каждому файлу на диске соответствует одно число: номер блока в котором этот файл начинается. На основе этой информации можно прочесть любой файл: из второй таблицы находим начальный блок файла и читаем его. Если файл на этом не кончается, находим из первой таблицы номер следующего блока и читаем его. Таким образом доходим до конца файла.

Если файловая система в порядке, то каждый блок на диске или свободен или дефектен или принадлежит в точности одному файлу. Также каждая последовательность блоков должна кончаться блоком для которого в FATе указано число -1. Если компьютер “зависнет”, то файловая система может испортиться. Написать программу, которая с целью проверки файловой системы вычисляет:

- a) число файлов, которые можно полностью прочесть (с начала до конца);
- b) максимальное число изначально читаемых файлов, которые могут оказаться нечитаемыми после удаления одного изначально читаемого файла (при удалении файла все блоки, принадлежащие ему, помечаются свободными).

Входные данные. На первой строке текстового файла `FATLID.SIS` число блоков n ($1 \leq n \leq 10000$) и число файлов m ($1 \leq m \leq 10000$). За этим следует одна пустая строка. На следующих n строках n целых чисел от -3 до $n-1$: содержание FATа. За этим следует одна пустая строка. На следующих m строках m целых чисел от 0 до $n-1$: номера начальных боков всех файлов.

Выходные данные. В текстовый файл `FATLID.VAL` вывести два целых числа: число файлов, которые в данном состоянии диска полностью читаемы и максимальное число читаемых файлов, которые могут оказаться нечитаемыми после удаления одного читаемого файла.

Пример. FAILID.SIS FAILID.VAL
 9 4 3 1

3
 8
 4
 -3
 6
 -2
 -1
 4
 -1

2
 0
 1
 7

5. КРОССВОРД

30 очков

10 секунд

Написать программу, которая заполняет кроссворд данными словами.

Входные данные. На первой строке текстового файла RIST.SIS три целых числа: число строк r ($3 \leq r \leq 30$) и столбцов v ($3 \leq v \leq 30$) в кроссворде и число слов n ($0 \leq n \leq 50$) в словаре. На следующих r строках расположен кроссворд, при чем точка означает белую и плюс чёрную клетку. На следующих n строках слова состоящие из заглавных латинских букв. В словаре указаны все присутствующие в кроссворде слова длиной не менее 2 букв.

Выходные данные. В текстовый файл RIST.VAL вывести r строк длиной v символов каждый: указанный во входных данных кроссворд, в строки и столбцы которого расставлены все слова из данного словаря. Слова можно вставлять в кроссворд только сверху вниз и слева направо. Кроме того каждое слово должно продолжаться от чёрной клетки или края кроссворда до другой чёрной клетки или другого края кроссворда. Известно, что задача имеет решение. Если решений несколько, вывести любое из них.

Пример. RIST.SIS RIST.VAL
 3 4 7 A+UT
 .+.. RIST
 UT+K
 ..+.
 ARU
 IT
 RIST
 TTK
 US
 UT
 UT