

Задача А. Экономный строитель

В стране Лилипутии грандиозное строительство. А вот места и материалов, как всегда, мало. Помогите архитектору Василию Тряпочкину подсчитать минимальное количество балок для образования каркаса нового дома. Дом строится на N квартир, каждая из которых представляет собой куб с ребром 1. Ребра куба – это балки. Каркас – это все балки, без стенок. Дом строится так, чтобы количество балок было минимальным.

На вход подается N - количество семей, нуждающихся в жилье (каждой семье соответствует один куб каркаса дома). Вывести количество балок.

Пример 1

input

1

output

12

Пример 2

input

4

output

33

Задача В. Период дроби

Определить период бесконечной периодической дроби, которая получается в результате деления целого числа A на целое число B . Требуется определить длину периода и сам период.

На вход подаются числа A и B . На выходе должно быть два числа: длина периода и собственно сам период. Для конечных дробей период и длина выводятся как 0. Для бесконечных непериодических дробей период и длина выводятся как -1.

Пример 1

input

1 65

output

6

153846

Пример 2

input

7 5

output

0
0

Пример 3

input

10 57

output

-1

-1

Задача С. Задача о вложенности слов

Дано множество из n уникальных слов $W = (w_1 \dots w_n)$. Строка $S = (s_1, \dots, s_m)$ составлена из m слов, s_i принадлежит W . Слова в строке S могут повторяться. Если дана последовательность $A X B Y A$, то слово B называется вложенным в слово A , если длины строк X и Y одинаковы (X , Y - это не слова, а строки из слов) и представляют собой зеркальное отражение друг друга. При этом X и Y не должны включать ни A , ни B , либо обе являются пустыми.

Также если дана последовательность $A X B Y B X A$, то B является вложенным словом в A .

Примеры:

External Inner External - Inner вложен в External.

Wrapper Layer Core Layer Wrapper - Layer вложен во Wrapper, а Core вложен в Layer, и соответственно Core вложен во Wrapper.

Shell Meet Perl Shell - ни Meet, ни Perl не являются вложенными в Shell.

На вход программы в первой строке подаётся последовательность слов, разделённых пробелами. В последующих строках подаются комбинации из двух и более слов, где каждое следующее слово вложено в предыдущее. Необходимо посчитать количество указанных вложений в исходной последовательности.

Признаком конца ввода является пустая строка.

Пример 1

input

a gg b c b gg a c a c a b b a

a b

a c

c a

output

1

3
1

Пример 2

input

stack item var item var ui stack
stack item
item var
ui stack

output

0
1
0

Пример 3

input

a b c d c b a b a c d c a c b c a b b b a c c
a b c
a c b
a b d
a d b
a b
b a

output

1
1
1
0
3
1

Задача D. О зависимостях

Имеется несколько файлов с исходным кодом программ. Каждый файл, для своей работы, может требовать использование другого файла. Необходимо определить порядок, в каком следует компилировать исходные файлы, чтобы все зависимости были разрешены. Если, после обработки нескольких файлов, все зависимости уже разрешены, и при этом ещё остались другие файлы — их зависимости также необходимо разрешить, подставляя новую последовательность зависимостей в конец уже полученной. Если файл, на который имеется ссылка, отсутствует в

приведённом списке исходных файлов — считать такой файл пустым (без зависимостей от других файлов).

Название каждого файла заканчивается двоеточием. После названия файла следуют названия файлов, от которых зависит текущий
Признаком конца ввода, является две пустые строки.

Пример 1

input

main:
stream
phone
user

libcall:

phone:
stream
libcall

user:
roles:

roles:

output

stream libcall phone roles user main

Пример 2

input
file5:
x_files

file1:
f1
f2
f3
f4

f4:
file5

f1:
f2

f3:

```
f2:  
f3  
123
```

```
123:  
yoyo
```

```
yoyo:
```

output

```
x_files file5 f3 yoyo 123 f2 f1 f4 file1
```

Задача Е. Повсюду кластеры

Во время физико-химического опыта был выращен кристалл. На поверхности кристалла осели частицы разных типов. Частицы одного типа, которые находятся рядом, образуют кластер.

Для анализа состава кластеров был создан снимок поверхности кристалла, представленный двумерным массивом символов, размером $M \times N$. Символ определяет тип частицы, которая осела на данном участке поверхности кристалла. В одной ячейке массива может находиться только одна частица.

Необходимо получить список типов частиц и для каждого типа определить количество и размер кластеров.

На входе в первой строке подаётся 2 числа: M и N . Далее следуют M строк по N символов - карта поверхности кристалла.

На выходе должны быть K строк (по количеству типов частиц). Каждая строка должна содержать тип частицы и затем через пробел список размеров кластеров, состоящих из частиц этого типа. Размеры кластеров нужно выдать по убыванию. Список частиц должен быть также отсортирован по общему количеству частиц данного типа по убыванию. Если количество разного типа символов одинаково - сортировать по номеру символа в ASCII.

Вход:

```
6 4  
**3**_  
*33*-*  
3*3*--  
***--*
```

Выход:

```
* 4 4 3 1 1
```

- 5 1
3 4 1

Вход:

14 7

...a....a..+..
..aaaa.++.++.
aaa..aa.+...+.
.a....a++.....
.a...+++.....
.aa+++.....++..
.....+...a.....

Выход:

. 35 9 8 5 5 1
+ 12 4 2
a 15 1 1

Задача F. Олимпиадные задачи

Близится день олимпиады, а задач маловато. Зброшена важная работа, под угрозой личная жизнь и финансовое благополучие, а члены оргкомитета, забыв про сон и отдых, создают задачи. И все потому, что списать задачи из интернета или из задачника им не позволяет совесть. Кто еще подает пример честного творческого труда, если не они? Усталость нарастает, воля ослабевает, и коварная память так и норовит подсунуть в условие новой задачи куски, идеи из каких-то учебников, сборников задач, турниров. Помогите совести – она не может логически осознать задачу, вместо этого ей нужно сравнивать тексты новых задач с обрывками памяти, и определить, является ли новая задача оригинальной, или это перелицованное старье.

Задача и обрывок памяти представляются в виде строк символов, например АААВВСА и АВВА. Для сравнения строки нужно выписать друг под другом.

АААВВСА

АВВА

Далее нужно вставить в обе строки некоторое количество пробелов (пробел будем обозначать черточкой) так, чтобы максимальное количество одинаковых букв оказалось друг под другом, т.е.

АААВВСА

А--ВВ-А

Возможны разные варианты таких вставок. Нужно выбрать

наилучший – который обладает максимальным баллом. За совпадение букв дается +10, за несовпадение -10, за пробел -5. Таким образом, оценка для данного примера ($10 - 5 - 5 + 10 + 10 - 5 + 10 = 25$) лучше, чем у конкурентов:

AAABBCA

A-B-B-A

($10 - 5 - 10 - 5 + 10 - 5 + 10 = 15$)

Если в итоге получилось, что количество одинаковых букв $\geq 50\%$ от минимума длин двух исходных строк $\min(4,7) = 4$, значит, строки похожи, задача похожа на воспоминание, и совесть отвергает такую задачу.

В первой строке входа число N – количество тестовых пар. Далее идут N строк, каждая содержит пару слов для сравнения. Вывести N строк с результатом сравнения: 1 – задача принимается, 0 – задача отвергается.

Примеры

Вход:

6

AAABBCA ABBA

YELLOW BLUE

WHITE LIGHT

МАМА ПАПА

BRYANSK KURSK

NOVOSIBIRSK SAMARA

Выход:

0

1

1

0

0

1

Задача G. День олимпиады

В одном вузе впервые провели олимпиаду по программированию. Всем очень понравилось, и олимпиаду решили повторить через год. Однако в этом году все оказалось не так просто: занятость членов оргкомитета сильно возросла, и все никак не удается выбрать подходящую дату: то Иванов не может, то Петров. Необходимо написать программу, которая выдаст подходящие даты проведения олимпиады в порядке возрастания, либо, если нет ни одной такой даты, выдаст 0.

На вход в первой строке подается число N – количество членов оргкомитета, далее идут N строк, в каждой из которых перечислены

даты, которые удобны i-му члену оргкомитета (даты не обязаны быть упорядоченными).

Примеры

input:

3

1.05.2012 7.05.2012 11.05.2012

2.05.2012 3.05.2012 7.05.2012

21.05.2012 7.05.2012

output:

7.5.2012

input:

3

1.05.2012 8.05.2012 11.05.2012

2.05.2012 3.05.2012 7.05.2012

4.05.2012 21.05.2012 6.05.2012

output:

0

input:

3

1.05.2012 8.05.2012 11.05.2012

2.05.2012 1.05.2012 8.05.2012

8.05.2012 21.05.2012 1.05.2012

output:

1.5.2012

8.5.2012