

Задача А. Кратчайшее расстояние

Входной файл: `dist.in`
 Выходной файл: `dist.out`
 Ограничение по времени: 2 секунды
 Ограничение памяти: 64 мегабайта

В этой задаче вам надо всего лишь найти расстояния от выделенной вершины до всех остальных вершин связного неориентированного графа, в котором не более 26 вершин.

Правда, сам граф задается слегка необычно. А именно, сначала задается дерево с выделенной вершиной, все вершины которого помечены символами от А до Z, а затем вершины с одинаковыми метками отождествляются.

Деревья указанного вида описываются следующим образом:

$\langle letter \rangle ::= A|B|\dots|Z$

$\langle list \rangle ::= \langle tree \rangle | \langle tree \rangle [, \langle list \rangle]$

$\langle forest \rangle ::= \langle tree \rangle | (\langle list \rangle)$

$\langle tree \rangle ::= \langle letter \rangle | (\langle tree \rangle) | \langle tree \rangle \rightarrow \langle forest \rangle$

Операция \rightarrow выполняется справа налево, то есть $A \rightarrow B \rightarrow C = A \rightarrow (B \rightarrow C)$. При этом любая буква α означает дерево с одной вершиной, помеченной символом α , а $x \rightarrow (y_1, y_2, \dots, y_s)$ означает дерево, полученное в результате соединения выделенной вершины дерева x с выделенными вершинами каждого из деревьев y_j ; выделенной вершиной в результирующем дереве является выделенная вершина дерева x .

Описание входного файла

На единственной строке входного файла содержится описание графа — строка из символов `A..Z`, `'-'`, `'>'`, `'<'`, `'&'`, `'('` и `')`. Строка не длиннее 255 символов.

Описание выходного файла

Для каждой из вершин данного графа следует выдать на отдельной строке метку этой вершины и расстояние от нее до выделенной вершины, разделенные пробелом.

Строки следует упорядочить в лексикографическом порядке.

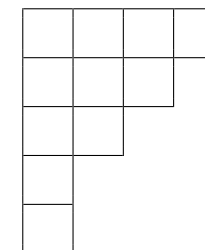
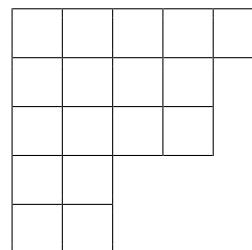
Пример

<code>dist.in</code>	<code>dist.out</code>
<code>R->(B->X->S->Y->Z,C->Z)</code>	B 1 C 1 R 0 S 3 X 2 Y 3 Z 2
<code>(A->B)->C</code>	A 0 B 1 C 1

Задача В. Построение таблицы Юнга

Входной файл: `young.in`
 Выходной файл: `young.out`
 Ограничение по времени: 2 секунды
 Ограничение памяти: 64 мегабайта

Зачастую вместо различных комбинаторных объектов бывает удобно рассматривать их графические представления. Для разбиений на слагаемые таким представлением является диаграмма Юнга. *Диаграммой Юнга* называется таблица из k строк, каждая из которых состоит из l_i клеток 1×1 , кроме того, для всех i таких, что $1 \leq i < k$, должно быть выполнено условие $l_i \geq l_{i+1}$. Все строки в диаграмме Юнга выровнены по левому краю. На рисунке изображены диаграммы Юнга, соответствующие разбиениям на слагаемые $5 + 4 + 4 + 2 + 2$ и $4 + 3 + 2 + 1 + 1$.



Иногда в комбинаторных задачах требуется рассматривать более сложный объект — пару из разбиения на слагаемые числа n и перестановки из n элементов. Связь между ними задается с помощью таблицы Юнга. *Таблицей Юнга* называется диаграмма

Юнга с n клетками, с расставленными в них числами от 1 до n таким образом, что числа каждой строки и каждого столбца образуют возрастающие последовательности.

Требуется вычислить количество способов построить таблицу Юнга по заданной перестановке n чисел. Соответствие таблицы Юнга перестановке определяется выписыванием из нее последовательно чисел первой строки (в порядке их следования в строке), затем второй, и т.д.

Описание входного файла

В первой строке входного файла задано число n ($1 \leq n \leq 1000$). Во второй строке содержатся n различных целых чисел от 1 до n , описывающих исходную перестановку.

Описание выходного файла

В выходной файл необходимо вывести количество способов построить по заданной перестановке таблицу Юнга.

Пример

young.in	young.out
4 1 3 2 4	2

Рисунок к примеру

1	3	1	3
2		2	4
4			

Задача С. Резисторы

Входной файл:	<i>нет</i>
Выходной файл:	<i>нет</i>
Ограничение по времени:	<i>нет</i>
Ограничение памяти:	<i>нет</i>

Радиолобитель Петя решил собрать детекторный приемник. Ему понадобился резистор сопротивления R Ом. В распоряжении Пети есть набор из n резисторов, сопротивления которых равны r_1, r_2, \dots, r_n соответственно. Петя помнит, как вычисляется сопротивление последовательного соединения двух резисторов $R_{new} = R_1 + R_2$ и параллельного соединения двух резисторов $R_{new} = \frac{R_1 \cdot R_2}{R_1 + R_2}$. Петя хочет спаять некоторую последовательно-параллельную схему из имеющегося набора резисторов такую, что ее

сопротивление ближе всего к искомому (то есть абсолютная величина разности значений минимальна).

Напомним, что схема, составленная из одного резистора, — последовательно-параллельная схема. Любая схема, полученная последовательным соединением двух последовательно-параллельных схем, — последовательно-параллельная, также любая схема, полученная параллельным соединением двух последовательно-параллельных схем, — последовательно-параллельная.

Разумеется, Петя не обязан использовать для изготовления схемы все резисторы.

Описание входного файла

Вам представлен набор из 15 тестов. Файлы, содержащие тесты, называются `resistors.##`. Таким образом, файлы имеют названия `resistors.01`, `resistors.02`, ..., `resistors.15`. В первой строке каждого входного файла заданы числа n, R . Во второй строке содержится последовательность сопротивлений имеющихся в наличии резисторов r_1, r_2, \dots, r_n . Значения всех сопротивлений могут быть вещественными числами. Для всех входных файлов $n < 10$.

Описание выходного файла

Выходные файлы следует называть `p27_<id>_c.##`, где `<id>` — ваш идентификатор. В выходной файл необходимо вывести сопротивление такой схемы, сопротивление которой меньше всего отличается от R . Результат выводите с шестью знаками после запятой. Следующие строки должны содержать инструкцию по сборке такой схемы. Каждая операция инструкции расположена на отдельной строке и имеет вид `<ID>=<ID1><ID2>...<IDk>` или `<ID>|<ID1><ID2>...<IDk>` ($k \geq 1$). Первая команда обозначает, что соединяются последовательно схемы с идентификаторами `<ID1><ID2>...<IDk>`, вторая — данный набор схем соединяется параллельно. Имеющийся набор резисторов соответствует схемам с идентификаторами `1, 2, ..., n`. В качестве идентификаторов составленных схем используйте заглавные латинские буквы. Последняя строка инструкции содержит единственный символ — идентификатор искомой схемы.

Пример

входной файл	выходной файл
3 1.66 1 2 1	1.666666 A 12 B=A3 B