

Задача А. Сумма делителей

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Заданы два натуральных числа a и b .

Необходимо найти сумму делителей числа a , не являющихся делителями числа b .

Формат входного файла

Первая строка входного файла содержит два записанных через пробел натуральных числа a и b ($1 \leq a, b \leq 10^6$).

Формат выходного файла

В выходной файл выведите единственное число — ответ на задачу.

Пример

<code>input.txt</code>	<code>output.txt</code>
6 4	9
3 2	3

Примечание

Рассмотрим первый пример. У числа 6 делители — это 1, 2, 3 и 6. У числа 4 делители — это 1, 2 и 4. Таким образом, ответ равен $3 + 6 = 9$.

Задача В. Последовательность

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Задана последовательность чисел a_1, a_2, \dots, a_n . При разбиении последовательности a по индексу i ($1 \leq i < n$) получаются две части: левая – включающая элементы от a_1 до a_i и правая – включающая элементы от a_{i+1} до a_n .

Назовем весом разбиения модуль разности сумм элементов левой и правой частей. Например, для последовательности $a = (-4, 5, 3, -3, 2)$ и $i = 2$ вес соответствующего разбиения равен $|(-4 + 5) - (3 + -3 + 2)| = |1 - 2| = |-1| = 1$.

Требуется найти количество разбиений, таких, что их вес наименьший.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($2 \leq n \leq 100$) — длину последовательности.

Вторая строка содержит n целых чисел a_i ($-1000 \leq a_i \leq 1000$) — элементы последовательности.

Формат выходного файла

В выходной файл выведите единственное число — ответ на задачу.

Пример

<code>input.txt</code>	<code>output.txt</code>
5 -4 5 3 -3 2	2
5 1 -1 1 3 -3	3
3 1 -2 1	2

Примечание

В первом примере, наименьший вес разбиения равен 1, а соответствующие значения i равны 2 и 4.

Задача С. Таблица

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Задана таблица, в каждой ячейке которой написана строчная буква латинского алфавита и шаблон слова, состоящий из строчных букв латинского алфавита и символов «?».

Шаблон можно прикладывать к таблице двумя способами: горизонтально (слева направо, от некоторой ячейки таблицы) и вертикально (сверху вниз от некоторой ячейки таблицы). При этом каждому символу шаблона будет соответствовать одна из букв таблицы. Назовем положение шаблона *корректным*, если шаблон целиком находится в пределах таблицы и соответствующие символы слова и шаблона совпадают. При этом считается, что символ «?» совпадает с любой буквой.

Требуется найти число корректных положений шаблона.

Формат входного файла

В первой строке входного файла содержатся два целых числа n и m ($1 \leq n, m \leq 300$) — количество строк и столбцов в таблице. В каждой из следующих n строк содержится m строчных латинских букв.

В следующей строке содержится целое число p ($2 \leq p \leq 300$) — длина шаблона. В следующей строке содержится строка p из строчных латинских букв и символов «?» — шаблон.

Формат выходного файла

В выходной файл выведите единственное число — количество корректных положений шаблона.

Примеры

input.txt	output.txt
3 4 abca caac ccbc 3 a?c	4
3 4 abca caac ccbc 3 ??c	6
2 2 ab cd 2 bd	1

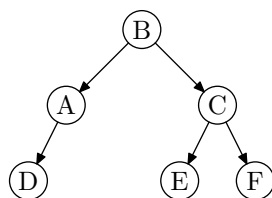
Задача D. Три обхода дерева

Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Рассмотрим двоичное дерево, в вершинах которого находятся различные прописные буквы латинского алфавита. Рассмотрим три обхода дерева:

LUR	левое-корень-правое (симметричный обход дерева)
ULR	корень-левое-правое (префиксный обход дерева)
LRU	левое-правое-корень (постфиксный обход дерева)

Выпишем буквы в вершинах в порядке посещения вершин при обходе, получив код дерева при обходе. Например, для дерева



коды, соответствующие обходам, имеют вид:

LUR	DABECF
ULR	BADCEF
LRU	DAEFCB

Вам задано дерево. Найдите коды обхода LUR, ULR и LRU.

Формат входного файла

Первая строка входного файла содержит n ($1 \leq n \leq 20$) — число вершин в дереве. В качестве пометок вершин используются первые n прописных букв латинского алфавита. На второй строке записано $2n$ символов: первая пара соответствует сыновьям вершины A, следующая пара — сыновьям вершины B и так далее. В каждой паре вначале указывается левый сын, затем — правый. Если соответствующего сына нет, вместо него указывается символ «-» (дефис).

Формат выходного файла

Выходной файл должен содержать коды обхода LUR, ULR и LRU заданного дерева.

Пример

<code>input.txt</code>	<code>output.txt</code>
6 D-ACEF-----	DABECF BADCEF DAEFCB