

Задача А. Преобразование Фурье

Имя входного файла: `fourier.in`
Имя выходного файла: `fourier.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Петя недавно нашел в книжке определение частичного преобразования Фурье набора комплексных чисел. Теперь он хочет, чтобы Вы написали ему программу для вычисления частичных преобразований Фурье, для того, чтобы получше изучить их свойства.

Вот определение частичного дискретного преобразования Фурье, найденное Петей в книжке:

Пусть $n = 2^s$, где $s \geq 0$ — целое число. Определим сначала инволюцию rev_s на множестве целых чисел от 0 до $2^s - 1$ следующим образом: $\text{rev}_0(0) = 0$, $\text{rev}_{s+1}(2x) = \text{rev}_s(x)$, $\text{rev}_{s+1}(2x + 1) = 2^s + \text{rev}_s(x)$ при $0 \leq x < 2^s$.

Зафиксируем теперь первообразный корень из единицы n -й степени $\zeta = e^{2\pi i/n} = \cos \frac{2\pi}{n} + i \sin \frac{2\pi}{n}$, и определим для любого t от 0 до s t -е частичное преобразование Фурье $\text{Four}_s^{(t)}(\mathbf{a})$ набора $\mathbf{a} = (a_0, a_1, \dots, a_{n-1})$ из n комплексных чисел следующим образом: $\text{Four}_s^{(t)}(\mathbf{a}) = \mathbf{b}^{(t)} = (b_0^{(t)}, b_1^{(t)}, \dots, b_{n-1}^{(t)})$, где

$$b_{2^{s-t}j+k}^{(t)} = \sum_{j'=0}^{2^t-1} \zeta^{2^{s-t}jj'} a_{2^{s-t}j'+\text{rev}_{s-t}(k)}$$

$$\text{при } 0 \leq j < 2^t, 0 \leq k < 2^{s-t}$$

Таким образом, $\text{Four}_s^{(s)} = \text{Four}_s$ — обычное преобразование Фурье:

$$b_j^{(s)} = \sum_{j'=0}^{n-1} \zeta^{jj'} a_{j'},$$

а $\text{Four}_s^{(0)}$ всего лишь переставляет числа в исходном наборе: $b_k^{(0)} = a_{\text{rev}_s(k)}$.

У Пети есть подозрение, что найти $\text{Four}_s^{(t)}$ очень просто, если уже известно $\text{Four}_s^{(t-1)}$, однако сам он вывести нужную формулу пока не сумел.

Ваша задача состоит в том, чтобы вычислить t -е частичное преобразование Фурье данного набора из 2^s комплексных чисел.

Формат входных данных

Первая строка входного файла содержит числа s и t , разделенные пробелом ($0 \leq t \leq s \leq 16$). Последующие 2^s строк содержат по два вещественных числа каждая — вещественную и мнимую часть соответствующего комплексного числа a_j .

Формат выходных данных

На каждой из 2^s строк выходного файла следует вывести два вещественных числа с шестью цифрами после запятой — вещественную и мнимую часть числа $b_j^{(t)}$.

Примеры

<code>fourier.in</code>	<code>fourier.out</code>
2 1	5.00000000 0.00000000
2.00 0.00	4.00000000 0.00000000
7.00 0.00	-1.00000000 0.00000000
3.00 0.00	10.00000000 0.00000000
-3.00 0.00	

Задача В. Пути в дереве

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 15 секунд
Ограничение по памяти: 512 мегабайт

Дано дерево из n вершин. Найдите для каждого d от 1 до $n - 1$ число путей длины d .

Формат входных данных

Первая строка содержит n — число вершин дерева ($1 \leq n \leq 50000$).

Следующие $n - 1$ строк содержат ребра дерева. Каждое ребро описывается парой чисел v_i, u_i — концы ребра ($1 \leq v_i, u_i \leq n$).

Формат выходных данных

Выведите $n - 1$ число: i -е число — число путей длины i .

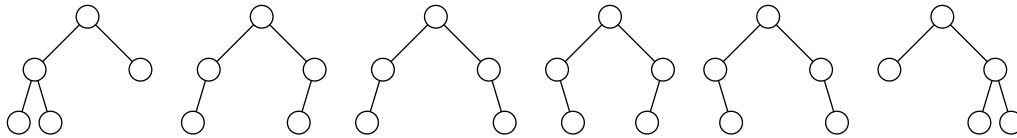
Примеры

стандартный ввод	стандартный вывод
3	2
1 2	1
2 3	
9	8
3 2	10
4 2	10
1 2	6
5 1	2
1 6	0
7 6	0
6 8	0
8 9	

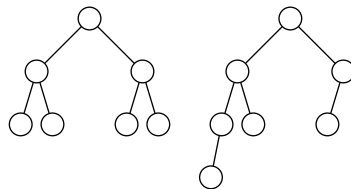
Задача С. AVL

Имя входного файла: `avl.in`
Имя выходного файла: `avl.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

AVL деревья, придуманные российскими учёными Адельсон-Вельским и Ландисом, являются примером сбалансированного бинарного дерева поиска. В терминологии AVL, подвешенное бинарное дерево называется сбалансированным, если для каждой вершины высоты её левого и правого поддеревьев отличаются не более, чем на один. Такое дерево, собственно, и называется AVL-деревом. Разумеется, существует далеко не единственное AVL-дерево при фиксированном числе вершин. К примеру, существует шесть AVL-деревьев с пятью вершинами, они изображены на рисунке ниже.



Деревья с одинаковым числом вершин могут иметь разную высоту, к примеру, на рисунке снизу нарисовано два дерева с семью вершинами, которые имеют высоты 2 и 3, соответственно.



Вам даны два числа — N и H , требуется найти число AVL-деревьев, которые состоят из N вершин и имеют высоту H . Поскольку их число довольно велико, выведите искомое количество по модулю 786 433.

Формат входных данных

Единственная строка входного файла содержит два числа — N и H ($1 \leq N \leq 65\,535$, $0 \leq H \leq 15$).

Формат выходных данных

Выведите единственное число — количество AVL деревьев с N вершинами высоты H , по модулю 786 433.

Примеры

<code>avl.in</code>	<code>avl.out</code>
7 3	16

Замечание

786 433 простое число, и $786\,433 = 3 \cdot 2^{18} + 1$.

Задача D. Дуэль

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт n деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Формат входных данных

Во входном файле содержится одна строка, состоящая из символов '0' и '1' — схема аллеи. Деревья обозначаются символом '1', кусты — символом '0'. Длина строки не превосходит 100 000 символов.

Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Примеры

<code>stdin</code>	<code>stdout</code>
101010101	4
101001	0

Замечание

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **101010101**, **101010101**, **101010101** и **101010101**.

Задача Е. Задача для второклассника

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вам даны два числа. Необходимо найти их произведение.

Формат входных данных

Входные данные состоят из двух строк, на каждой из которых находится целое одно **целое** число, длина которого не превосходит двухсот пятидесяти тысяч символов.

Формат выходных данных

Выведите произведение данных чисел.

Примеры

<code>stdin</code>	<code>stdout</code>
2	4
2	

Задача F. HEX-Hell и сломанная строка

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Серёжа потерял место, где в редакторе шестнадцатеричных кодов HEX-Hell находилась его строка. И, так как он большой молодец, саму строку он сломал.

Напоминаем, что один байт — две шестнадцатеричных цифры из диапазона `[0-9A-F]`. Последовательность кодов в редакторе в данный момент имеет длину не более 125 000 байт (т.е. в ней не более 250 000 символов, и она имеет четную длину).

А также, у Вас есть серёжина битовая строка из нулей и единиц, «поломанная» в некоторых местах (некоторые биты будут заменены на знаки ?).

Пожалуйста, для каждого возможного начала этой строки в редакторе выведите количество совпадающих нулей и единиц. При этом можно считать, что знаки вопроса совпадают с чем угодно! По этой информации горе-Серёжа разберется со своими строками сам. Только найдите количество совпадений!

Формат входных данных

Во входном файле две непустых строки. Первая состоит из символов от 0 до 9 и от A до F. Количество таких символов чётно и не более 250 000.

Вторая строка состоит из нулей, единиц и знаков вопроса. Её длина кратна восьми (она тоже задаёт последовательность байт) и не превосходит четырех длин первой строки (она помещается в редактор).

Формат выходных данных

Если длина (в символах) первой строки n , а второй k , то Вам необходимо вывести в первой строке выходного файла $n/2 - k/8 + 1$ целых чисел — количество совпадающих бит при прикладывании второй строки к некоторому месту первой строки.

Прикладывания упорядочены естественным образом — слева направо.

Примеры

<code>stdin</code>	<code>stdout</code>
B2D6 1011?010	8 5

Замечание

B2D6 в двоичном виде это 10110010 11010110

При прикладывании шаблона 1011?010 к первому байту 10110010 получаем восемь совпадений, а ко второму 11010110 — пять.

Задача G. Системы счисления

Имя входного файла: `radix.in`
Имя выходного файла: `radix.out`
Ограничение по времени: 0.75 секунд
Ограничение по памяти: 256 мегабайт

Переведите число из системы счисления a в систему счисления b .

Формат входных данных

В первой строке задана начальная система счисления a ($2 \leq a \leq 10$). Во второй строке задано само число, его длина не превышает 100 000 символов. В третьей строке записана система счисления b , в которую нужно перевести число ($2 \leq b \leq 10$).

Формат выходных данных

В единственной строке выведите заданное число в системе счисления b .

Примеры

<code>radix.in</code>	<code>radix.out</code>
3	27
1000	
10	