

Задача А. Украденный массив

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	0.8 секунд
Ограничение по памяти:	256 мегабайт

Мальчик Петя любит массивы. Недавно ему подарили огромный массив чисел F размера 2^n . Петя человек странный, и при виде массива из чисел сразу начинает считать какие-то суммы на нём. Специально для этого он купил в магазине новый пустой массив P размера 2^n и начал его заполнять по следующему правилу: $P[i] = \sum_{j \& i = j} F[j]$. Другими словами для каждого j , такого что j — подмаска i (т.е. побитовое «И» чисел i и j равно j), Петя прибавил $F[j]$ к изначально нулевому значению $P[i]$.

Но потом случилось ужасное — массив F украли! Теперь Петя хочет разыскать массив F , но он не помнит, какие значения там были изначально. Единственное что у него есть — массив P . Помогите Пете восстановить массив F .

Формат входных данных

В первой строке входных данных дано одно число n ($1 \leq n \leq 20$).

В следующей строке даны 2^n чисел, i -е из них — значение $P[i]$ ($-10^9 \leq P[i] \leq 10^9$) (нумерация ведётся с нуля).

Формат выходных данных

В одной строке выведите 2^n чисел, i -е из которых — значение $F[i]$ (нумерация ведётся с нуля).

Примеры

стандартный ввод	стандартный вывод
2 1 2 3 4	1 1 2 0
3 1 3 4 10 6 14 16 36	1 2 3 4 5 6 7 8

Задача В. Cake Tasting

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мегабайт

Карина очень любит пирожные. В этом месяце Карине удалось посетить n различных кондитерских, предлагающих дегустацию своих изделий. Для каждой кондитерской Карина записывала, какие виды пирожных она попробовала, в специальное приложение на своем смартфоне. Но после всех дегустаций Карина обнаружила, что приложение показывает только количество различных видов пирожных, которые она попробовала. Причем эту информацию можно посмотреть для любого из $2^n - 1$ подмножества кондитерских. Карина очень расстроена и считает, что даже эти данные приложение вычислило неверно.

Помогите Карине понять, верны ли данные, и, если верны, предъявите любой подходящий вариант записей о дегустациях.

Формат входных данных

В первой строке задано число n ($1 \leq n \leq 19$) — количество кондитерских. В следующей строке содержатся $2^n - 1$ чисел a_i ($1 \leq i \leq 2^n - 1$; $1 \leq a_i \leq 1000$). Число a_i равно количеству различных видов пирожных, которые Карина пробовала в кондитерских с номерами j , где j -й разряд двоичного представления числа i равен единице. Например, если S_k — множество видов пирожных в k -й кондитерской, то $a_1 = |S_1|$, $a_2 = |S_2|$, $a_3 = |S_1 \cup S_2|$, $a_4 = |S_3|$, и так далее.

Формат выходных данных

Если данные приложения неверны, выведите в единственной строке слово «No». Иначе, выведите в первой строке слово «Yes», а в следующих n строках выведите возможный вариант записей Карины. Для каждой кондитерской запись должна начинаться с числа k_i ($1 \leq k_i \leq 1000$), равного количеству видов пирожных в i -й кондитерской. Затем, в той же строке, выведите k_i различных чисел $s_{i,j}$, не превышающих 10^9 по модулю, — виды пирожных.

Если существует несколько возможных вариантов записей, выведите любой. Гарантируется, что если существует решение, то существует также и решение, в котором все $k_i \leq 1000$.

Примеры

стандартный ввод	стандартный вывод
2 2 3 4	Yes 2 1 4 3 1 2 3
2 2 2 5	No
3 3 2 4 3 4 4 5	Yes 3 1 2 5 2 1 4 3 1 2 3

Замечание

В первом примере есть две кондитерские. В первой кондитерской Карина попробовала 2 вида пирожных, во второй — 3, в первой и второй вместе — 4. Один из возможных вариантов, соответствующих этим числам, такой — в первой кондитерской Карина попробовала пирожные 1 и 4 вида, во второй — 1, 2 и 3 вида, тогда в первой и второй вместе она попробовала пирожные четырех видов: 1, 2, 3 и 4.

Во втором примере в первой и второй кондитерской Карина попробовала по 2 вида пирожных, а в сумме — 5 видов. Такого очевидно не могло быть.

В третьем примере есть три кондитерские. Один из возможных ответов такой:

$$\begin{aligned}a_1 &= |S_1| = |\{1, 2, 5\}|, \\a_2 &= |S_2| = |\{1, 4\}|, \\a_3 &= |S_1 \cup S_2| = |\{1, 2, 4, 5\}|, \\a_4 &= |S_3| = |\{1, 2, 3\}|, \\a_5 &= |S_1 \cup S_3| = |\{1, 2, 3, 5\}|, \\a_6 &= |S_2 \cup S_3| = |\{1, 2, 3, 4\}|, \\a_7 &= |S_1 \cup S_2 \cup S_3| = |\{1, 2, 3, 4, 5\}|.\end{aligned}$$

Задача С. Задача для второклассника

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2.5 секунд
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

стандартный ввод	стандартный вывод
2 2	4
1 -1	-1
-1 -239	239
0 -1	0

Задача D. ФФТ по известному модулю

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Предлагается в этой задаче реализовать перемножение двух многочленов по известному модулю $MOD = 998\,244\,353$. Используя написанную вами в этой задаче функцию, вы сможете сдать без проблем большинство задач по комбинаторике, в которой нужно ФФТ.

Формат входных данных

В первой строке находится целое число n ($0 \leq n \leq 18$).

Во второй строке находится 2^n целых чисел $a_0, a_1, \dots, a_{2^n-1}$ ($0 \leq a_i < MOD - 1$). Первый многочлен для перемножения это $A(x) = \sum_{i=0}^{2^n-1} a_i x^i$.

Во второй строке находится 2^n целых чисел $b_0, b_1, \dots, b_{2^n-1}$ ($0 \leq b_i < MOD - 1$). Первый многочлен для перемножения это $B(x) = \sum_{i=0}^{2^n-1} b_i x^i$.

Формат выходных данных

Пусть многочлен $C(x) = A(x) \cdot B(x)$. Все коэффициенты при перемножении берутся по модулю MOD . Тогда напишем, что $C(x) = \sum_{i=0}^{2^{n+1}-1} c_i x^i$. Выведите $c_0, c_1, \dots, c_{2^{n+1}-1}$.

Пример

стандартный ввод	стандартный вывод
1	1 2 1 0
1 1	
1 1	

Задача E. HEX-Hell и сломанная строка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 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.

Пример

стандартный ввод	стандартный вывод
B2D6 1011?010	8 5

Замечание

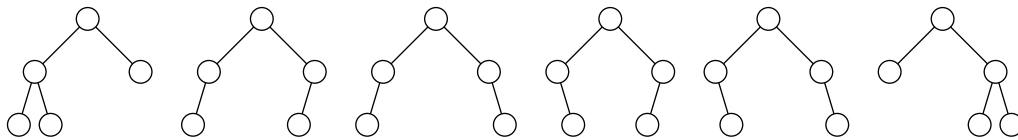
B2D6 в двоичном виде это 10110010 11010110

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

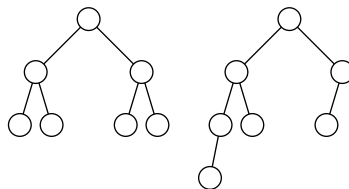
Задача F. AVL

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
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.

Пример

стандартный ввод	стандартный вывод
7 3	16

Замечание

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

Задача G. Количество путей

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 5 секунд
Ограничение по памяти: 256 мегабайт

Дано дерево из n вершин. Для каждого $d = 1 \dots n - 1$ найдите количество путей длины d .

Формат входных данных

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

Следующие $n - 1$ строк содержат пары чисел u_i, v_i ($1 \leq u_i, v_i \leq n$), описывающие рёбра дерева.

Формат выходных данных

Выведите $n - 1$ число, где i -е — количество путей длины i .

Пример

стандартный ввод	стандартный вывод
3	2
1 2	1
2 3	

Задача Н. Перемножаем по модулю

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Кате подарили валентинку, на которой было написано n чисел a_i . Разумеется, Катя сразу же решила посчитать $\sum_{i < j} a_i \cdot a_j$. Поскольку Катя хотела, чтобы ответ получился маленьким, но не очень,

Катя решила брать все произведения по модулю $P = 200'003$, а сумму по модулю **не брать**.

Таким образом, от вас требуется посчитать $\sum_{i < j} ((a_i \cdot a_j) \pmod{P})$. Удачи!

Формат входных данных

В первой строке вводится число n ($2 \leq n \leq 200'000$) — количество чисел на валентинке.

В следующей строке через пробел вводится n чисел a_i ($0 \leq a_i \leq P - 1$) — последовательность чисел с валентинке.

Формат выходных данных

Выведите одно число — $\sum_{i < j} ((a_i \cdot a_j) \pmod{P})$.

Примеры

стандартный ввод	стандартный вывод
4 2019 0 2020 200002	474287
5 1 1 2 2 100000	600013

Задача I. Эпическая свёртка

Имя входного файла: стандартный ввод
 Имя выходного файла: стандартный вывод
 Ограничение по времени: 4 секунды
 Ограничение по памяти: 256 мегабайт

Вам даны два массива a_0, a_1, \dots, a_{n-1} и b_0, b_1, \dots, b_{m-1} , и целое число c .

Вычислите следующую сумму:

$$\sum_{i=0}^{n-1} \sum_{j=0}^{m-1} a_i b_j c^{i^2 j^3}$$

Так как это число может быть ну очень большим, выведите его по модулю 490019.

Формат входных данных

Первая строка содержит три целых числа n , m и c ($1 \leq n, m \leq 100\,000$, $1 \leq c < 490019$).

Следующая строка содержит ровно n целых чисел a_i и задаёт массив a ($0 \leq a_i \leq 1000$).

Последняя строка содержит ровно m целых чисел b_i и задаёт массив b ($0 \leq b_i \leq 1000$).

Формат выходных данных

Выведите одно число — значение суммы по модулю 490019.

Примеры

стандартный ввод	стандартный вывод
2 2 3 0 1 0 1	3
3 4 1 1 1 1 1 1 1 1	12
2 3 3 1 2 3 4 5	65652

Замечание

В первом примере единственное ненулевое слагаемое соответствует $i = 1$, $j = 1$ и равно $1 \cdot 1 \cdot 3^1 = 3$.

Во втором примере все слагаемые равны 1.