

Задача А. Схема

Имя входного файла: *нет*
Имя выходного файла: *нет*
Ограничение по времени: *нет*
Ограничение по памяти: *нет*

Булевой функцией называется функция из $\{0,1\}^n$ в $\{0,1\}$. Булеву функцию можно задать *таблицей истинности* — значениями функции на области определения. Например, можно рассмотреть функции «И», «ИЛИ» и «НЕ», которые мы будем обозначать \wedge , \vee и \neg соответственно. Выпишем их таблицы истинности:

x	y	$x \wedge y$	$x \vee y$	$\neg x$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Булеву функцию f можно задать *схемой*. Схему можно определить так. Пусть имеется n булевых переменных x_1, x_2, \dots, x_n , называемых *входами*. Пусть также имеется некоторое число булевых переменных y_1, y_2, \dots, y_m , называемых *проводниками*. Пусть для каждого проводника схемы задана функция «И», «ИЛИ» или «НЕ», выражающая его значение через другие проводники и входы. При этом требуется, чтобы среди зависимостей не было циклических. Кроме того, среди проводников выделен один, который называется *выходом*. Его значение является функцией от x_1, x_2, \dots, x_n , которая нас интересует. Размером схемы называется число m (количество проводников).

Вам задана булева функция f . Нужно построить схему как можно меньшего размера, которая вычисляет f . Отметим, что достигать оптимального размера не требуется.

Формат входного файла

Входные файлы к этой задаче являются открытыми; их можно найти по адресу <http://192.168.21.239/circuit/>.

Первая строка входных данных содержит n — количество переменных. Следующая строка содержит 2^n значений функции на входах, упорядоченных лексикографически (например, для $n = 2$ значения будут записаны в таком порядке: $f(0,0)$, $f(0,1)$, $f(1,0)$ и $f(1,1)$).

Формат выходного файла

Решением задачи считается набор выходных файлов, соответствующих входным, упакованный архиватором `gzip` и утилитой `tar` (команда `'tar -czf archive.tar.gz ??out'` создаёт `gzip`-архив `'archive.tar.gz'`, содержащий все файлы, подходящие под шаблон `'??out'`). Архив должен содержать по одному файлу `XX.out` на каждый входной файл `XX` (без расширения) или на любую часть этих файлов, и не должен содержать никаких других файлов и папок.

Каждый выходной файл должен содержать описание найденной вами схемы для соответствующей функции. Первая строка должна содержать размер схемы s . Далее в s строках должны содержаться описания проводников. В последней из этих s строк должно содержаться описание выхода схемы.

Теперь объясним, как должно выглядеть описание проводника. Описание должно начинаться с операции, которая соответствует данному проводнику. «НЕ» обозначается `!`, «И» обозначается `&`, а «ИЛИ» — `|`. Далее в строке должны находиться аргументы операции: один для «НЕ» и два для «ИЛИ» и «И». Каждый аргумент соответствует либо проводнику, либо входу. В первом случае вы должны выводить номер проводника, во втором — номер входа, умноженный на -1 . Номера проводников, от которых зависит данный, должны быть строго меньше его номера (для гарантии отсутствия циклических зависимостей).

Проводники и входы нумеруются с единицы.

При отсылке производится проверка на соответствие всех файлов в архиве формату вывода, и результат сообщается участнику. Независимо от этого результата, оценка решения участника производится по последнему посланному архиву с ответами.

Система оценки

Оценка за решение — это сумма оценок за каждый тест. Оценка за отдельный тест вычисляется следующим образом. Максимальный балл за тест, который можно получить, равен 10. Если ваша схема некорректная, не вычисляет требуемую функцию или содержит больше 1 000 000 проводников, то вы получите 0 баллов.

В таблице 1 указано количество баллов, которое получит схема в зависимости от количества проводников в ней. В первом столбце записано количество баллов, во втором и третьем — минимальное и максимальное количество проводников, соответственно.

Таблица 1: Система оценки

Балл	от	до
10	0	50 000
9	50 001	60 000
8	60 001	80 000
7	80 001	100 000
6	100 001	150 000
5	150 001	400 000
4	400 001	600 000
3	600 001	700 000
2	700 001	800 000
1	800 001	1 000 000
0	1 000 001	∞

Пример

Пример входного файла	Пример выходного файла
3 01101001	14 ! -1 ! -2 ! -3 & 1 2 & 4 -3 & 1 -2 & 6 3 & -1 2 & 8 3 & -1 -2 & 10 -3 5 7 12 9 13 11

Задача В. Числа

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

Даны N целых неотрицательных чисел. Мы рассматриваем их попарные суммы (все N^2 сумм). Для каждого двоичного разряда от 0 до K нужно посчитать, сколько раз в этих суммах в нем стояла единица.

Формат входного файла

Числа N, K, P, Q ($0 \leq N \leq 10^7$, $1 \leq K \leq 24$, $1 \leq P, Q \leq 10^9 + 7$, P и Q — простые). Последовательность чисел a_0, \dots, a_{N-1} генерируется так. $a_0 = 1, a_{i+1} = (a_i \cdot P + Q) \bmod 2^K$

Формат выходного файла

В первой строке выведите $K+1$ целое число — количества единиц в разрядах (Можно было бы вывести остальные количества, не только первые $K+1$. Но они нули.) Первым выводите нужно ответ для младшего разряда.

Примеры

стандартный ввод	стандартный вывод
3 10 3 5	4 4 4 5 4 3 0 0 0 0
10 3 3 5	50 25 48 16
100 10 17 239	5000 5000 5002 5002 4998 5014 4989 5029 4985 5015 4188

Замечание

Решение, работающее для $K \leq 20$ и $N \leq 10^6$, будет набирать не менее 70 баллов.

Задача С. Палиндромы

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

Сотрудник НИИ Исследования Данных Строк (НИИИДС) Вася Пупкин изучает свойства палиндромов. Напомним, что строка s длины n называется палиндромом, если она равна s^R (развёрнутой строке s). Вася занялся подсчётом количества подстрок данной строки, являющихся палиндромами. С этой задачей

Вася справился достаточно быстро. Теперь Вася решил поочерёдно заменять символы в строке на символ ‘?’, который Вася считает равным любому символу. То есть, например, Вася считает равными строки “ABA” и “A?A”, “ABB” и “AB?”, а, соответственно, строки “AB?”, “ABC??”, “?BCB?” он считает палиндромами. Вася хочет после каждой замены символа на ‘?’ определять количество подстрок строки s , являющихся палиндромами (в том смысле, в котором это понимает Вася). При этом подстроки Вася считает различными, если у них различаются позиции начала и/или конца. Вам предложено автоматизировать Васиные действия, чтобы он смог заняться чем-нибудь ещё.

Отметим, что до тех пор, пока вы не выведете очередной ответ, информация об очередной операции со строкой доступна не будет.

Формат входного файла

Первая строка входных данных содержит строку s , состоящую только из строчных латинских букв. Гарантируется, что строка s непуста, и её длина n не превосходит 4 000. Далее во входных данных содержится n чисел от 1 до n — номера символов, заменяемых на ‘?’.

После прочтения строки s выведите количество её подстрок, являющихся палиндромами.

Далее Вы должны n раз проделать следующую последовательность действий:

- считать число k — номер символа строки, заменённого на ‘?’ на текущем шаге ($1 \leq k \leq n$),
- подсчитать количество подстрок текущей строки, являющихся палиндромами,
- вывести это число на отдельной строке,
- выполнить команду `flush(output)` в Паскале и `fflush(stdout)` в C и C++.

Гарантируется, что никакой символ исходной строки не будет дважды заменён на ‘?’.

Не рекомендуется использовать потоки ввода/вывода C++ (`fstream`), так как они работают существенно медленней `FILE *` и могут стать причиной превышения ограничения по времени.

Формат выходного файла

Смотрите описание входных данных.

Пример

стандартный ввод	стандартный вывод
abac	5
3	7
2	9
1	10
4	10

Замечание

Комментарий к примеру: В строке `abac` палиндромами являются подстроки $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(1,3)$. После замены третьего символа на ? получаем строку “ab?c”, в которой палиндромами являются подстроки $(1,1)$, $(2,2)$, $(3,3)$, $(4,4)$, $(1,3)$, $(2,3)$, $(3,4)$. После замены в ней второго символа получаем строку “a??c”, в которой палиндромом не является только подстрока $(1,4)$. После следующих замен все подстроки будут являться палиндромами.

Задача D. T-пути

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

Задан неориентированный невзвешенный связный граф $G = (V, E)$ с выделенным множеством вершин $T \subset V$. T -путь — простой путь, начинающийся и заканчивающийся в различных вершинах множества T . Найдите длину кратчайшего T -пути в графе, а также остаток количества различных кратчайших T -путей по модулю 1 000 000 007.

Формат входного файла

В первой строке записаны числа n, m, t ($2 \leq n \leq 10^5$, $1 \leq m \leq 5 \cdot 10^5$, $2 \leq t \leq n$) — количество вершин, рёбер графа и размер множества T соответственно. На второй строке записаны t различных чисел — элементы множества T . На следующих m строках записаны пары чисел x, y — номера вершин, соединённых ребром ($x \neq y$, $1 \leq x, y \leq n$). В графе нет петель и кратных рёбер.

Формат выходного файла

Выведите два числа — длину кратчайшего T -пути и количество различных кратчайших T -путей по модулю 1 000 000 007.

Система оценки

Каждый тест оценивается в некоторое количество баллов V . За каждое пра-

Летние учебно-тренировочные сборы кандидатов в сборную России по информатике
Малоярославец, вторник, 30 июня 2009 года

вильно выведенное число Вы получите $\frac{V}{2}$ баллов.

Примеры

стандартный ввод	стандартный вывод
4 4 2 1 3 1 2 2 3 3 4 4 1	2 2
5 5 3 5 1 3 1 2 2 3 5 2 3 4 4 1	2 4