

Задача A. Numbers

Имя входного файла: `numbers.in`
Имя выходного файла: `numbers.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Найти количество таких перестановок набора попарно различных целых чисел S , в которых любые два соседние числа в перестановке имеют различный остаток при делении на заданное целое положительное число M .

Формат входного файла

В первой строке входного файла заданы два целых числа: N — количество элементов в множестве S ($1 \leq N \leq 200$) и используемое при построении перестановок число M ($1 \leq M \leq 1000$). Во второй строке заданы элементы последовательности S — N неотрицательных целых чисел, не превосходящих 10^6 .

Формат выходного файла

Выведите одно число — количество соответствующих перестановок, взятое по модулю $10^9 + 3$.

Пример

<code>numbers.in</code>	<code>numbers.out</code>
4 6 2 4 6 8	12
3 300 30 321 112	6

Задача В. Ranking

Имя входного файла: `ranking.in`
Имя выходного файла: `ranking.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

На игровом сервере «Чараев Gaming Emptiness» прошёл однокруговой турнир по игре в «Чапаева»: каждый из N участников сыграл с каждым одну партию. При этом в каждой партии обязательно был выявлен победитель.

Администрация сервера хочет упорядочить игроков по их результатам так, чтобы для любых двух игроков A и B , где игрок A стоит в списке выше игрока B , выполнялось хотя бы одно из двух условий:

- Игрок A выиграл у игрока B .
- Существует такой игрок C , стоящий в списке между игроками A и B , который проиграл игроку A и выиграл у игрока B .

Ваша задача — построить такое упорядочение.

Формат входного файла

В первой строке входного файла задано целое число N ($1 \leq N \leq 500$) — количество участников турнира. В последующих N строках задана таблица турнира. В i -й строке заданы N символов — результаты i -го участника. j -й символ этой строки равен 0, если i -й участник проиграл j -му, и 1, если i -й участник выиграл у j -го (при этом i -м символом строки является 0).

Формат выходного файла

В выходной файл выведите одну строку — упорядочение всех N участников, соответствующее требованиям администрации сервера. Участники нумеруются от 1 до N в порядке, в котором они перечислены во входном файле. В случае, если такое упорядочение невозможно, выведите `'impossible'`.

Пример

ranking.in	ranking.out
4 0000 1010 1001 1100	2 3 4 1
5 00110 10100 00010 01000 11110	5 2 1 3 4

Задача C. Roads

Имя входного файла: `roads.in`
Имя выходного файла: `roads.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

В небольшой торговой республике N городов, причём некоторые из них соединены дорогами с двусторонним движением. При этом из любого города можно проехать в любой другой по сети дорог, два города может соединять более одной дороги, а также могут существовать «кольцевые» дороги, соединяющие город сам с собой, однако для каждой пары (возможно, совпадающих) городов существует не более 4 дорог, непосредственно соединяющих эти города.

Парламент принял решение заасфальтировать некоторые дороги так, чтобы из любого города можно было проехать в любой другой по заасфальтированным дорогам и при этом суммарная длина заасфальтированных дорог была бы минимальна. Но при выборе конкретного маршрута в парламенте возникла дискуссия, и ни один проект не получил большинства.

Перед следующим голосованием спикеру парламента нужно узнать, сколько существует различных способов реализации решения парламента. Решение этой задачи поручено Вам.

Формат входного файла

В первой строке входного файла заданы два числа — количество городов N ($1 \leq N \leq 5 \cdot 10^4$) и количество дорог M ($1 \leq M \leq 10^5$). В последующих M строках заданы дороги — сначала идут два номера a_i и b_i ($1 \leq a_i, b_i \leq N$) городов, которые соединяет дорога, а затем — длина дороги l ($1 \leq l \leq 2^{30}$).

Формат выходного файла

Выведите одно число — количество различных способов заасфальтировать дороги в соответствии с решением парламента, взятое по модулю $10^6 + 3$.

Пример

<code>roads.in</code>	<code>roads.out</code>
3 5 2 1 7 2 1 7 3 2 7 1 3 7 3 3 9	5

Задача D. Text typing

Имя входного файла: `text.in`
Имя выходного файла: `text.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Во время соревнований «Facepalm Hackers Cup» сотрудник Facepalm Вася установил на компьютере хакера Пети сканер клавиатуры, записавший в файл коды всех нажатых хакером Петей клавиш (при этом Петя, в частности, набирал пароль). Но, так как Вася был сотрудником Facepalm, то сканер был написан с ошибкой и записал набранные символы в случайном порядке.

По полученному Васей файлу для каждого из предполагаемых паролей вычислите, мог ли этот пароль использоваться Петей.

Формат входного файла

В первой строке входного файла в некотором порядке перечислены символы, которые набирал хакер Петя. Среди символов могут встречаться только заглавные латинские буквы, строка не пуста и длина строки не превосходит 200.

Во второй строке входного файла задано число N ($1 \leq N \leq 100$) — количество предполагаемых паролей. В каждой из последующих N строк заданы предполагаемые пароли — непустые последовательности из не более, чем 100 заглавных латинских букв.

Формат выходного файла

Для каждого пароля в порядке следования паролей во входном файле выведите ‘YES’, если у Пети мог быть данный пароль, и ‘NO’, если из полученного Васей лога следует, что такой пароль невозможен.

Пример

<code>text.in</code>	<code>text.out</code>
FACEPALM	YES
4	YES
MACE	NO
A	NO
FF	
AXE	

Задача E. UFO

Имя входного файла: `ufo.in`
Имя выходного файла: `ufo.out`
Ограничение по времени: 2 seconds
Ограничение по памяти: 64 Mebibytes

Недалеко от прямоугольного огорода совершил посадку НЛО, который выжег вокруг себя круг заданного диаметра. По заданным координатам точки посадки НЛО вычислите площадь той части огорода, которая была повреждена при посадке.

Формат входного файла

Во входном файле заданы 7 целых чисел: координаты (x_g, y_g) точки посадки НЛО, диаметр круга d ($1 \leq d \leq 1000$), а также параметры прямоугольника, заданного системой неравенств $x_1 \leq x \leq x_2$, $y_1 \leq y \leq y_2$, в порядке $x_1 x_2 y_1 y_2$. Все координаты не превосходят 1000 по абсолютной величине.

Формат выходного файла

Выведите одно число — площадь повреждённой части огорода с точностью до 7 знаков.

Пример

<code>ufo.in</code>	<code>ufo.out</code>
4 2 6 4 9 3 7	4.1251038
3 2 4 -3 0 -3 2	0.0000000

Задача F. Worlds

Имя входного файла:	<code>worlds.in</code>
Имя выходного файла:	<code>worlds.out</code>
Ограничение по времени:	2 seconds
Ограничение по памяти:	64 Mebibytes

При объединении нескольких виртуальных трёхмерных миров в один возникла необходимость перевести координаты, заданные в различных системах счисления, в двоичную систему счисления. Для каждого из исходных миров выбрана «ключевая точка» и заданы её координаты. При этом особенности представления данных таковы, что:

- Максимальное значение каждой из координат использует на один знак в соответствующей системе счисления больше, чем соответствующая координата «ключевой точки» данного виртуального мира и является целой степенью основания соответствующей системы счисления. Например, если используется восьмеричная система и координаты ключевой точки равны $(1000_8, 77_8, 123_8)$, то максимальное значение координаты x равно 10000_8 , максимальное значение координаты y — 100_8 , максимальное значение координаты z — 1000_8 .
- Все координаты — целые неотрицательные числа.
- Основание системы счисления может принимать значения в интервале от 0 до 4096. При этом если основание системы счисления не превосходит 36, то одна цифра представляется во входном файле одним символом (0-9,A-Z), если не превосходит 36^2 , то одна цифра представляется двумя символами (36-ричной записью соответствующей цифры с ведущими нулями), например, 1599_{10} в 40-ичной системе счисления представляется как 1313_{40} , в оставшихся случаях представляется тремя символами (строятся аналогично двухсимвольным цифрам, например, 1610_{10} представляется как $100A_{1600}$). Ведущие нули, даже являющиеся частью цифры, в записи опускаются.

Требуется выяснить, какое минимальное двоичных разрядов необходимо отвести под каждую из трёх координат для того, чтобы после преобразования можно было представить координаты любой (в том числе и максимальной) точки в любом из виртуальных миров.

Формат входного файла

В первой строке входного файла задано в десятичной записи одно число N ($1 \leq N \leq 10^4$) — количество исходных виртуальных миров. Каждая из последующих N строк начинается с одного числа b_i ($1 \leq b_i \leq 4096$), записанного в десятичной системе счисления, за которым следуют три «числа», записанных в системе счисления с основанием b_i — соответственно x , y и z -координаты соответствующей ключевой точки. Координаты заданы в формате, описанном в условии задачи, при этом каждая из координат не может быть задана более, чем 2^{20} символами и не превосходит 10^{10^9} . Общий размер входного файла при этом не превосходит 2 мегабайт.

Формат выходного файла

В выходной файл выведите три числа, записанных в десятичной системе — минимальное количество двоичных разрядов, требуемое для записи каждой из координат.

Пример

worlds.in	worlds.out
2 5 14 5 3 8 767 74 4444	10 7 13
1 10 0 0 0	4 4 4
2 40 12345ABCDEF ABCDEFGH SNWS 2000 THE SAMPLE 1001	32 22 22