

Задача 1. Футбол

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

Марсианскому садовнику Казимиру Казимировичу нравятся земные виды спорта: он регулярно смотрит матчи по телевизору, изучает результаты прошедших матчей. Особенно нравится садовнику футбол. Он старается не пропустить ни одно важное соревнование. Один раз Казимир Казимирович даже слетал на Землю, чтобы посмотреть футбольный матч вживую.

Однажды садовник увидел в газете результаты всех матчей футбольного чемпионата и заинтересовался, кто же стал победителем.

Чемпионат проходил по следующим правилам. Каждая команда играла с каждой по два раза: первый раз игра проходила на стадионе первой команды, а второй раз — на стадионе второй команды. За победу в одном матче команде присуждалось три очка, за ничью — одно очко, а за поражение не присуждалось ничего. Считается, что команда выиграла матч, если забила больше голов, чем соперник. Победителем чемпионата становилась та команда, которая набрала наибольшее количество очков за все матчи. Для определенности будем считать, что если команд, набравших наибольшее число очков, несколько, то победителем среди них считается команда с лексикографически наименьшим названием. (Строка s лексикографически меньше, чем строка t , если s должна стоять в словаре раньше, чем t)

Казимир Казимирович сейчас занят высадкой редкого вида дерева — двоичной сбалансированной березы. Поэтому он попросил Вас написать программу, которая сможет определить победителя чемпионата по известным результатам матчей. Может оказаться и так, что список матчей в газете был неполным, потому что чемпионат еще не закончился. В таком случае Казимиру Казимировичу интересно, сколько матчей еще осталось сыграть.

Формат входных данных

В первой строке находятся два целых числа n и m ($1 \leq n \leq 100$, $1 \leq m \leq n \cdot (n - 1)$) — количество команд и количество матчей соответственно.

Во второй строке через пробел записаны n различных названий команд, от 1-ой по n -ую. Названия всех команд содержат от 1 до 20 латинских букв, причем первая буква названия заглавная, а остальные — строчные.

В каждой из следующих m строк содержится две последовательности из букв латинского алфавита a , b , а также два целых числа r_a и r_b ($a \neq b$, $0 \leq r_a, r_b \leq 99$) — названия команд, которые играли в матче и количество голов, которые забила каждая из них. Первой записана та команда, на стадионе которой проходил матч. Гарантируется, что команды a и b присутствуют в списке команд s_i .

Гарантируется, что каждая пара команд (a, b) (с учетом порядка) играла не более одного матча.

Формат выходных данных

Если данные о матчах полные (т. е. все команды сыграли друг с другом по два матча), то выведите в первой строке выходных данных строку «YES», а во второй строке — название команды-победителя.

Иначе в первой строке выходных данных выведите «NO». Во второй строке выходных данных выведите целое число k — количество матчей, которые еще предстоит сыграть.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n = 2$	3	
2	$n \leq 10$	14	1
3	$m = n \cdot (n - 1)$	8	
4	$m = 0$	8	
5	Нет дополнительных ограничений	67	1 - 4

Примеры

стандартный ввод	стандартный вывод
3 4 Minsk Mogilev Brest Mogilev Minsk 1 2 Brest Minsk 2 0 Minsk Mogilev 3 3 Mogilev Brest 1 0	NO 2
3 6 Minsk Mogilev Brest Mogilev Minsk 1 2 Brest Minsk 2 0 Minsk Mogilev 3 3 Mogilev Brest 1 0 Minsk Brest 3 1 Brest Mogilev 2 5	YES Minsk

Замечание

Во втором примере две команды (Minsk и Mogilev) набирают по семь очков. Но поскольку название Minsk лексикографически меньше, то именно эта команда объявляется победителем.

Задача 2. Странные антинейтрино

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

Марсианские ученые разрабатывают новый двигатель для межгалактических полетов. Двигатели на нейтринной тяге уже достигли своего предела по скорости, поэтому исследования в этом направлении были прекращены. После длительных и сложных экспериментов ученые обратили свой взгляд на антинейтрино. Они считают, что двигатель на основе энергии этих элементарных частиц будет более быстрым и энергоэффективным, чем существующие нейтринные двигатели.

Чтобы построить мощный двигатель, необходимо изучить свойства антинейтрино. В распоряжении ученых имеется n антинейтрино, каждый из которых имеет *странность* a_i — их квантовую характеристику.

Антинейтрино проявляют особые свойства, когда выстроены в ряд. В этом случае можно вычислить *странность* ряда из этих элементарных частиц. Если считать, что в ряд выстроены антинейтрино со странностью a_1, a_2, \dots, a_n , то тогда странность ряда равна

$$S(a) = \max(a_1 + a_2, a_2 + a_3, \dots, a_{n-1} + a_n).$$

Для построения эффективного двигателя необходимо использовать ряды антинейтрино с небольшой странностью. Поэтому ученым необходимо разместить все имеющиеся у них антинейтрино в ряд таким образом, чтобы минимизировать странность полученного ряда. Вам было поручено помочь ученым и написать программу, которая разместит эти элементарные частицы оптимальным образом.

Формат входных данных

В первой строке входных данных находится целое число n ($2 \leq n \leq 3 \cdot 10^5$) — количество антинейтрино в распоряжении ученых.

Во второй строке входных данных находится n целых чисел a_i ($1 \leq a_i \leq 10^9$) — странность i -го антинейтрино.

Формат выходных данных

В первой строке выходных данных выведите целое число S — минимально возможная странность, если расположить все n антинейтрино в ряд оптимальным способом.

Во второй строке выходных данных выведите n целых чисел b_i — оптимальное расположение антинейтрино в ряд. Последовательность b_i должна быть получена из a_i путем перестановки элементов.

Если способов расположить антинейтрино в ряд оптимальным образом несколько, то выведите любой из них.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$n = 2$	3	
2	$n \leq 10$	7	1
3	$n \leq 15$	9	1, 2
4	$n \leq 5000$	33	1 - 3
5	$a_i \leq 2$	9	
6	Нет дополнительных ограничений	39	1 - 4

Примеры

стандартный ввод	стандартный вывод
4 1 2 3 4	5 4 1 2 3
7 2 2 2 2 2 1 1	4 2 1 2 1 2 2 2

Задача 3. Быстрые вычисления

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

Мальчик Петя и его друг, робот Petya++, любят играть друг с другом в разные игры. Однажды Петя выписал на доске все целые числа от l до r подряд и попросил Petya++ назвать количество подотрезков, на которых исключающее ИЛИ всех чисел равно в точности s . Иными словами, Petya++ должен был найти количество таких пар i, j , что $l \leq i \leq j \leq r$ и $i \oplus (i+1) \oplus \dots \oplus j = s$, где \oplus обозначает операцию побитового исключающего ИЛИ.

Petya++ справился с этой задачей практически мгновенно. Петя спросил у своего друга, как ему удалось так быстро найти ответ, но тот не стал раскрывать свой секрет. Поэтому мальчик обратился за помощью к Вам и попросил написать программу, которая позволила бы ему решать эту задачу так же быстро, как и Petya++. Поскольку количество требуемых подотрезков может быть очень большим, необходимо найти лишь его остаток от деления на число $10^9 + 7$.

Формат входных данных

В единственной строке входных данных находится три целых числа l, r и s ($1 \leq l \leq r \leq 10^{18}$, $0 \leq s \leq 10^{18}$) — минимальное и максимальное число, которое Петя выписал на доске, а также требуемое значение исключающего ИЛИ на подотрезке.

Формат выходных данных

Выведите одно целое число — количество подотрезков, на которых исключающее ИЛИ всех чисел равно в точности s . Ответ необходимо вывести по модулю $10^9 + 7$.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$l, r \leq 3$	3	
2	$r - l \leq 80$	9	1
3	$r - l \leq 2000$	9	1, 2
4	$r - l \leq 3 \cdot 10^5$	13	1 - 3
5	$l = 1, s = 0$	8	
6	$s = 0$	8	
7	Нет дополнительных ограничений	50	1 - 6

Пример

стандартный ввод	стандартный вывод
1 8 0	4

Замечание

Исключающее ИЛИ — операция, которая на вход принимает два бита и возвращает бит 0, если биты на входе равны и 1 в противном случае. Например: $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. Для применения побитового исключающего ИЛИ двух чисел эти числа сначала переводят в двоичную систему счисления, а затем применяют исключающее ИЛИ к каждому из разрядов. Например, $6 \oplus 3 = 5$, поскольку $6 = 110_2$, $3 = 11_2$. Применив исключающее ИЛИ поразрядно, получаем $5 = 101_2$:

$$\begin{array}{r} 110_2 \\ \oplus 11_2 \\ \hline 101_2 \end{array}$$

В языке программирования Pascal побитовое исключающее ИЛИ чисел a и b обозначается $a \text{ xor } b$, а в языках Python и C++ — $a \wedge b$.

Теперь рассмотрим пример из условия.

В нем подходят четыре подотрезка:

- $1 \oplus 2 \oplus 3 = 0$
- $1 \oplus 2 \oplus 3 \oplus 4 \oplus 5 \oplus 6 \oplus 7 = 0$
- $2 \oplus 3 \oplus 4 \oplus 5 = 0$
- $4 \oplus 5 \oplus 6 \oplus 7 = 0$

Задача 4. Глобальная сеть

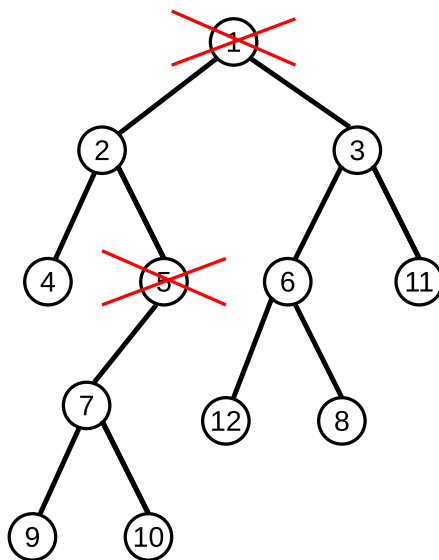
Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Когда-то давно правительство Венеры решило обеспечить всех граждан простым и доступным способом связи между любыми двумя точками планеты. Для этого была построена глобальная сеть *Венеранет*, состоящая из n маршрутизаторов и $n - 1$ кабеля. В то далекое время, когда сеть проектировалась, прокладка кабелей по всей планете была дорогой и трудозатратной. По этой причине сеть построена таким образом, что между любыми двумя маршрутизаторами существует ровно один простой маршрут, проходящий по кабелям. Иными словами, сеть имеет древовидную структуру.

Главная проблема такой структуры сети — ее ненадежность. Если по какой-либо причине один из маршрутизаторов выйдет из строя, то через него нельзя будет передавать информацию. В таком случае сеть может стать несвязной, т. е. может оказаться так, что между двумя маршрутизаторами невозможно передать информацию.

Правительство планирует в ближайшее время увеличить надежность сети Венеранет. Но для начала необходимо оценить устойчивость текущей структуры сети к сбоям. Пусть вышли из строя некоторые два маршрутизатора u и v ($u \neq v$). Тогда вся сеть распадется на некоторое количество *подсетей*, т. е. подмножеств исходной сети, внутри которых существует маршрут от каждого компьютера к каждому.

В качестве примера рассмотрим сеть на рисунке ниже:



Предположим, что в данной сети вышли из строя маршрутизаторы с номерами 1 и 5. Тогда вся сеть распадется на три подсети: одна будет содержать маршрутизаторы 2 и 4, другая — компьютеры 7, 9 и 10, а третья — маршрутизаторы 3, 6, 8, 11 и 12. Заметим, что между маршрутизаторами из разных подсетей невозможно передать информацию.

Пусть размеры подсетей — a_1, a_2, \dots, a_k соответственно. Тогда величину $C = a_1^2 + a_2^2 + \dots + a_k^2$ назовем *связностью* сети.

Для оценки надежности сети необходимо вычислить ее связность в худшем или лучшем случае, т. е. минимально или максимально возможную связность в случае выхода из строя ровно двух маршрутизаторов. Однако из-за огромного размера сети Венеранет вычисление этой характеристики оказалось очень трудозатратным. Поэтому Вам было поручено написать программу, которая сможет быстро оценить надежность сети.

Формат входных данных

В первой строке входных данных дано число t ($1 \leq t \leq 2$). В случае, если $t = 1$ требуется найти

максимально возможную связность в случае выхода из строя ровно двух маршрутизаторов; в случае $t = 2$ требуется найти минимально возможную связность.

В второй строке входных данных находится целое число n ($3 \leq n \leq 10^5$) — количество маршрутизаторов в сети Венеранет.

В каждой из следующих $n - 1$ строк находится два целых числа a_i и b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — номера маршрутизаторов, которые соединены i -м кабелем.

Гарантируется, что описанная во входных данных сеть является связной и имеет древовидную структуру.

Формат выходных данных

Выведите одно целое число — ответ на задачу.

Система оценки

№	Дополнительные ограничения	Баллы за подзадачу	Необходимые подзадачи
1	$t = 1, n \leq 10$	3	
2	$t = 1, n \leq 300$	4	1
3	$t = 1, n \leq 3000$	4	1, 2
4	$t = 1$	4	1 - 3
5	$t = 2, n \leq 300$	9	
6	$t = 2, n \leq 3000$	10	5
7	Нет дополнительных ограничений	66	1 - 6

Примеры

стандартный ввод	стандартный вывод
2 12 1 2 1 3 2 4 2 5 3 6 3 11 5 7 6 12 6 8 7 9 7 10	28
1 4 1 2 2 3 3 4	4

Замечание

В первом примере рассмотрена сеть, которая изображена на рисунке выше. Если в данной сети выйдут из строя маршрутизаторы 3 и 5, то образуются четыре подсети: $\{1, 2, 4\}$, $\{6, 8, 12\}$, $\{7, 9, 10\}$ и $\{11\}$. Их размеры равны 3, 3, 3 и 1 соответственно, поэтому связность сети Венеранет в данном случае будет равна $C = 3^2 + 3^2 + 3^2 + 1^2 = 28$. Отметим, что существуют и другие способы добиться связности 28, например, вывести из строя маршрутизаторы 2 и 3.