Задача A. Стеки

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

В давние времена, когда Магнето и Чарльз Ксавьер работали в одной команде, они любили играть в одну игру. Правила победы, очередность ходов и всё остальное их не интересовало — кто убедительнее докажет, что сейчас его ход, тот и ходит. Определение победителя тоже выливалось в философский спор, не имеющий отношения к игре, и переходящий к вопросу о роли мутантов в обществе.

Нас же интересует сам процесс игры. У игроков есть n изначально пустых стеков. Каждый игрок может сделать один из трёх ходов:

- A l r x — положить на вершину каждого из стеков с l по r число x;
- G x — спросить у соперника число, лежащее на вершине x-го стека;
- R i — отменить i-й по порядку запрос добавления числа на стеки: удалить соответствующее число из каждого стека.

В один день Чарльз заметил, что Магнето стал отвечать слишком быстро. После непродолжительного наблюдения, он заметил, что хитрый Магнето написал программу, которая делает всё за него. Чарльзу это, конечно, не понравилось, но он не стал обвинять соперника в мошенничестве: это бы ещё больше увеличило напряжение в их отношениях. Вместо этого он решил сам автоматизировать процесс, чтобы не отставать от своего извечного соперника.

Напишите программу, которая сумеет играть в такую игру не хуже, чем Ксавьер и Магнето.

Формат входного файла

В первой строке даны числа n ($1 \le n \le 10^5$) — число стеков, и m ($1 \le m \le 10^5$) — число запросов.

В следующих m строках даны запросы в формате, описанном в условии. Гарантируется, что все номера стеков лежат в интервале от 1 до n, а числа, которые кладутся на стек, удовлетворяют ограничению ($1 \le x \le 10^9$). Для любого запроса на отмену добавления гарантируется, что соответствующий запрос добавления уже был исполнен, и ни один запрос не отменяется дважды.

Формат выходного файла

Для каждого запроса второго типа выведите одно число — число, находящееся на вершине соответствующего стека, или -1, если соответствующий стек пуст.

Пример

stacks.in	stacks.out
3 7 A 1 3 5 A 2 3 3 G 1 G 2 R 1 G 1 G 2	5 3 -1 3

Задача B. Геном

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

Боливар Траск изучает геном мутантов в своей лаборатории, чтобы сделать своих Стражей ещё более совершенными.

На геном мутанта влияют гены `X`, `Y` и `Z`. Они взаимодействуют между собой следующим образом: каждый раз под действием катализатора все гены раздваиваются (каждый ген делится на два новых гена), после этого половина сформированных генов `X` и половина `Y` взаимойдействуют, чтобы получить ген `Z`. Вторая половина `X` и половина `Z` формируют ген `Y`. И оставшиеся половины `Y` и `Z` производят ген `X`. Таким образом, если геном состоял из x_0, y_0, z_0 соответсвующих генов, то после одного применения катализатора он будет состоять из x_1, y_1, z_1 генов, где $x_1 = y_0 + z_0$, $y_1 = x_0 + z_0$ и $z_1 = x_0 + y_0$.

Траск исследует отношение генов `X` и `Y`. Для исследований ему нужно знать, на сколько гена `X` будет больше, чем гена `Y` после k применений катализатора. Ваша задача — написать программу, которая рассчитает $x_k - y_k$, где x_k и y_k — количество генов `X` и `Y` соответственно после k применений катализатора.

Формат входного файла

Во единственной строке входного файла дано четыре числа x, y, z и k ($0 \le x, y, z, k \le 10^{18}$) — первоначальное количество генов `X`, `Y` и `Z` соответсвенно, а также количество применений катализатора.

Формат выходного файла

В единственной строке выходного файла выведите, на сколько гена `X` больше, чем гена `Y` после k применений катализатора.

Пример

genom.in	genom.out
1 1 1 4	0
4 3 2 1	-1

Комментарий

Во втором примере после одного применения катализатора будет 5 генов `X` и 6 генов `Y`. Соотвественно, гена `X` на -1 больше, чем гена `Y`.

Задача C. Пароль

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

Это интерактивная задача.

Недавно Профессор Ксавьер с помощью своего Церебро смог узнать о готовящемся наступлении Магнето. Но этого мало, нужна полная информация о размере его войск, времени и месте первого удара. Взломав систему безопасности на компьютере Магнето, Профессор нашел на нём нужный файл. Но чтобы открыть этот файл, нужно знать пароль, который знает только сам Магнето.

К счастью, Профессор нашёл на компьютере программу, которая используется для восстановления пароля. Этой программе можно подавать на вход строку произвольной длины, тогда на выходе программа выдаст сообщение о том, входит ли введённая строка в строку-пароль как подстрока. Ксавьеру также удалось узнать длину пароля и то, что пароль хранится в бинарном виде — он состоит только из нулей и единиц.

Теперь он хочет сделать несколько запросов к программе и полностью узнать пароль. Но у него не очень много времени — довольно скоро Магнето поймет, что его компьютер взломан, и поменяет все пароли. Чарльз точно знает, что успеет сделать 1024 запроса к программе. Помогите ему — скажите, какие запросы надо делать.

Протокол взаимодействия с программой жюри:

В самом начале программа жюри сообщает вашей программе натуральное число n ($1 \le n \le 1000$) — длину строки-пароля.

Дальше во время взаимодействия вашей программы с программой жюри несколько раз повторяются следующие действия:

- ваша программа сообщает программе жюри непустую строку, состоящую из символов `0` и `1`, которую вы хотите проверить на принадлежность паролю
- программа жюри сообщает вашей программе:
  - «`Success`», если ваша строка и есть пароль
  - «`0`», если в пароле нет такой подстроки
  - «`1`», если в пароле есть такая подстрока
- в случае, если вы нашли пароль, вам необходимо завершить работу своей программы
- в противном случае, описанные действия начинают повторяться сначала

Гарантируется, что пароль состоит только символов `0` и `1`, а также имеет длину ровно n.

Пример

stdin	stdout
3 0 0 0 0 0 Success	000 001 010 011 100 101

Комментарий

Для корректной работы программы после каждой операции вывода данных вам необходимо делать следующие операции:

- В языке Pascal: `flush(output);`
- В C/C++: `fflush(stdout);`
- В Java: `System.out.flush();`
- В Python: `sys.stdout.flush();`

Кроме этого, не забывайте после каждой выведенной строки ставить перевод строки.

Задача D. Забег

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

Команда Людей Икс решила выяснить, кто из них самый быстрый бегун. Для этого они организовали трассу, состоящую из контрольных точек. Трасса представлена в виде неориентированого графа с n вершинами, обозначающими контрольные точки, и m рёбрами. Для каждого ребра известна его длина.

По правилам забега каждый участник должен пробежать k контрольных точек так, чтобы последовательность вершин, соответствующих им, представляла простой путь. Однако Профессор Икс выяснил, что в системе, регистрирующей участника на контрольной точке, есть ошибка. Для каждого участника система запоминает только последнюю посещённую контрольную точку. Профессор Икс решил схитрить — он решил составить не кратчайший простой путь, а такой маршрут минимальной длины из k контрольных точек, что каждые две последовательные точки различны, и между ними есть ребро.

Формат входного файла

В первой строке входного файла дано два числа n и m ($1 \le n \le 100000, 1 \le m \le 200000$) — количество вершин и количество ребер в графе. В следующей строке дано целое число k ($2 \le k \le 100000$) — длина маршрута. В i-й из следующих m строк дана тройка чисел a_i, b_i, w_i ($1 \le a_i, b_i \le n, 1 \le w_i \le 10000$) — описание i-го ребра. Гарантируется, что в графе нет петель.

Формат выходного файла

Выведите минимальную длину маршрута.

Пример

run.in	run.out
3 3 3 1 2 1 2 3 1 1 3 2	2

Задача E. Задача

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

Во время обучения в Институте для одарённых подростков у Росомахи часто возникали проблемы с дисциплинами Профессора Ксавьера. Строгий лектор с мощнейшим интеллектом требовал от своих студентов неимоверной сообразительности. Часто он придумывал задачи, чтобы проверить их умственные навыки. Одна из задач была такая: Из набора, содержащего n чисел, требовалось выбрать подмножество размером k, чтобы разница между максимальным и минимальным числами из этого подмножества была минимальна. Росомахе очень не хотелось решать эту задачу в голове, поэтому он решил воспользоваться компьютером и написать программу. Недавно, просматривая архив своих программ, Росомаха обнаружил ошибку в реализации этой задачи. На переписывание программы у Росомахи нет времени, поэтому он просит вас помочь. Не стоит отказывать ему в этом, ведь к кому вы обратитесь, когда на планету в очередной раз нападёт Магнето?

Формат входного файла

В первой строке входного файла даны два числа n и k ($2 \le k \le n \le 100000$) — количество чисел в наборе и размер множества, которое надо выбрать. В следующей строке дано n чисел a_i ($0 \le a_i \le 10^9$).

Формат выходного файла

В единственной строке выходного файла выведите через пробел k чисел из исходного набора, таких, что разница максимального и минимального из них минимальна.

Пример

task.in	task.out
5 4 3 0 5 1 2	0 1 2 3
7 3 8 8 9 13 1 5 10	8 8 9

Задача F. Росомаха и стеллаж

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

Зверю на день рождения подарили стеллаж с книгами. Стеллаж имеет форму дерева, где вершинам соответствуют полки с книгами, а рёбрам — соединения полок между собой. В каждой вершине находится несколько (возможно, ноль) книг.

Это не простое дерево, оно обладает следующими свойствами: у каждой вершины не более двух детей, и количество книг в каждой из них равно суммарному количеству книг в её детях (кроме листьев — про листья известно, что в них не более одной книги).

Зверь был очень рад этому подарку, поэтому сразу повесил его на стену. Но Росомаха, проходя мимо, случайно зацепил дерево своими адамантиевыми когтями, и оно упало на пол. Росомаха собрал все книги, которые нашёл, но некоторых явно не хватало, поэтому он решил взять несколько из своей коллекции и добавить их незаметно на полки, чтобы получившееся дерево обладало изначальными свойствами. Но после того, как он это сделал, стало только хуже. Теперь надо всё исправить и всё-таки вернуть дереву его изначальные свойства — чтобы количество книг в каждой вершине, кроме листьев, равнялось суммарному количеству книг в детях, а в листьях было не более одной книги. Зверь скоро вернётся, поэтому Росомахе надо действовать как можно быстрее.

Немного подумав, он понял, что не всегда выгодно только добавлять книги на полки, иногда выгодно их убирать оттуда. За одну секунду Росомаха может либо положить на какую-то полку книгу, либо забрать её с какой-то из полок. Ваша же задача заключается в следующем: помогите Росомахе найти наименьшее время, за которое он сможет получить такими операциями дерево, обладающее теми же свойствами, что и дерево, подаренное изначально Зверю на день рождения.

Формат входного файла

В первой строке входного файла дано число n — количество вершин в дереве ($1 \le n \le 5000$). Во второй строке входного файла даны n целых чисел a_i ($0 \le a_i \le 5000$) — количество книг на i-й полке. В i-й из следующих $n-1$ строк даны два числа a, b ($1 \le a, b \le n$) — ребро дерева.

Корень дерева находится в вершине с номером 1.

Можно считать, что в коллекции Росомахи бесконечное число книг.

Формат выходного файла

В единственной строке выходного файла выведите минимальное количество секунд, которое понадобится Росомахе, чтобы скрыть свою оплошность.

Пример

tree.in	tree.out
2 1 0 1 2	1
5 5 1 3 0 1 1 2 1 3 3 4 3 5	4

Задача G. Логины

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

Профессор Ксавьер хочет начать сезон командных олимпиад по программированию в школе для одарённых мутантов. Каждой команде выдан логин, но так как большинство мутантов в школе ещё дети, многие школьники неправильно запомнили свой логин.

Правильный логин представляет из себя строку, состоящую из двух частей. Первая часть — это строка из двух букв «io», вторая часть — набор цифр. Таким образом логин «io182865» является корректным, а логин «ind3038» — нет.

Каждый школьник хочет узнать, корректен ли его логин. Чтобы не смотреть каждый такой запрос детально, Профессор просит вас написать программу, которая сможет определить корректность логина.

Формат входного файла

В единственной строке входного файла дана строка s ($1 \le |s| \le 10$), состоящая из строчных латинских букв и цифр — логин участника.

Формат выходного файла

В единственной строке выходного файла выведите `Correct`, если логин корректный для данного контеста. Иначе выведите `Incorrect`.

Пример

logins.in	logins.out
io1234	Correct
ind3038	Incorrect

Задача H. Нечетное или четное?

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

Несмотря на то, что Джубили является полноправным членом команды Людей Икс, она всё же школьница и обязана делать домашнее задание.

Джина Грей задала ей на дом непростое задание. Нужно определить чётность огромного арифметического выражения с двумя неизвестными x и y, если известна чётность этих переменных.

К сожалению, обязанности супергероя не оставили Джубили ни сил, ни времени. Она просит вас написать программу, которая определит чётность данного ей выражения.

Формат входного файла

В самой первой строке входного файла дана строка s ($1 \le |s| \le 10^6$) — выражение, содержащее в себе цифры, знаки `+`, `-`, `*`, а также переменные `x` и `y`. Во второй и третьей строках входного файла содержится описание чётности переменных x и y в следующем формате: `Odd`, если значение переменной нечетно и `Even` в противном случае. Гарантируется, что во входной строке нет двух подряд стоящих знаков арифметических операций, что строка начинается с цифры или переменной, все числа без ведущих нулей, а также что слева и справа от переменной стоят арифметические знаки, начало или конец строки.

Формат выходного файла

В единственной строке выходного файла выведите `Odd`, если выражение нечётно, и `Even`, если нет.

Пример

oddoreven.in	oddoreven.out
1+23+x+456*y-7 Odd Even	Even
x*x*x*y-x Odd Even	Odd