

Задача А. Продвинутый будильник

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Последнее время Поликарп стал часто пропускать сигналы своего будильника, всего лишь нужно было ввести комбинацию 4 8 15 16 23 42 и можно дальше сладко спать. К сожалению, он стал часто просыпаться важные дела и решил покончить с этим.

На днях Поликарп купил себе новый будильник, представляющий из себя хитрое устройство, которое просто так не выключишь. Чтобы будильник перестал работать нужно ввести десятичное число, состоящее из цифр 0 и 1, которое будет делиться нацело на все свои циклические сдвиги. Циклический сдвиг числа — это число, полученное переносом первых нескольких цифр в конец. Например, для числа 4020 это будут числа 4020, 204, 2040 и 402.

Всё было бы так просто, если бы будильник не требовал каждый раз новую комбинацию. Поэтому Поликарп решил вводить каждое утро числа по порядку, начиная с числа 1. Но, он беспокоится, что у него когда-нибудь не получится ввести нужное число, поэтому Поликарп решил узнать K -ое по порядку число заранее. Помогите ему, он в долгу не останется.

Формат входных данных

В единственной строке ввода содержится натуральное число $1 \leq K \leq 10^5$ — номер интересующего числа для Поликарпа.

Формат выходных данных

Выведите искомое число без лидирующих нулей.

Примеры

стандартный ввод	стандартный вывод
2	10
5	111

Задача В. Пароль

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

Поликарп забыл свой пароль, но он помнит, что пароль является десятичным числом. В записи числа присутствуют только ненулевые цифры, причём все они различны. Чтобы усложнить пароль, Поликарп выбрал среди всех вариантов самое большое чётное число. Помогите Поликарпу вспомнить пароль, если известно, что его длина равна n .

Формат входных данных

В единственной строке ввода содержится целое число $1 \leq n \leq 9$.

Формат выходных данных

Выведите пароль, который не может вспомнить Поликарп.

Примеры

стандартный ввод	стандартный вывод
1	8
2	98

Задача С. ContestCalendar

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

Поликарп недавно принял участие в одном престижном турнире по программированию но, к сожалению не смог в нём выиграть. Осознав свою ошибку — недостаток опыта в олимпиадах, он решил потренироваться до следующего сезона. Просмотрев десятки сайтов и форумов по олимпиадному программированию, он подготовил список соревнований и онлайн тренировок. Про каждое соревнование Поликарп записал название и даты начала и конца.

Когда Поликарп изучал форумы, он заметил, что все соревнования можно оценить по интересности. Он понял, что интересность олимпиады напрямую зависит от количества буквы «С» в аббревиатуре названия этой олимпиады. Если её совсем нет, Поликарп считает это соревнование скучным. Если присутствует хотя бы одна буква «С», то соревнование вполне интересное, а если их больше одной, то соревнование считается действительно достойным. Так как название олимпиады состоит из набора слов, то аббревиатура строится следующим образом: для каждого слова в названии определяется, из каких букв оно состоит, если только из заглавных, то слово полностью входит в аббревиатуру, иначе входит только первая буква слова в заглавной форме. Например, для названия «Russian AI cup» аббревиатурой будет «RAIC».

Помогите Поликарпу, по имеющейся информации о соревнованиях и текущей дате определите, какие соревнования ещё не закончились (считается, что соревнование не закончилось, если дата его окончания позже текущей даты), а так же определите интересность каждого соревнования. Для большей подробности прочитайте формат вывода.

Формат входных данных

В первой строке ввода содержится текущая дата. Во второй строке содержится число N — количество соревнований ($1 \leq N \leq 100$).

Далее в $3N$ строках идут описания соревнований. В первой строке каждого описания содержится название, во второй — дата начала и в третьей — дата окончания соревнования. Все даты описаны в формате «DD.MM.YYYY hh:mm», где D — день, M — месяц, Y — год (от 0 до 9999), h — час (от 0 до 23), m — минута (от 0 до 59).

Месяц и день заданы корректно. Дата начала всегда раньше даты конца. Все названия соревнований различны. Название состоит только из букв латинского алфавита, цифр и пробелов, длина названия не превосходит 50 символов. Каждое слово в названии состоит либо только из цифр, либо только из букв. Слова разделены ровно одним пробелом.

Формат выходных данных

Выведите все соревнования, которые проходят на данный момент или начнутся в будущем, относительно заданной текущей даты. Соревнования в выводе следует отсортировать по возрастанию даты начала, а если начала равны, то по названию соревнования. Каждое соревнование следует вывести в отдельной строке в формате `status name (type)`, где `status` — это «Running», если соревнование идёт в данный момент или «Coming soon», если оно начнётся в будущем. `name` — это название соревнования, которое дано во входных данных, а `type` — это каким для себя Поликарп считает это соревнование: «boring» (если соревнование кажется ему скучным), «interesting» (если соревнование вполне интересное) или «must coding!» (если соревнование действительно достойное).

Примеры

стандартный ввод	стандартный вывод
19.04.2014 10:00 3 Abakan programming contest 19.04.2014 10:00 19.04.2014 14:00 ACM ICPC World Finals 2013 03.07.2013 09:30 03.07.2013 15:00 ACM ICPC World Finals 2014 25.06.2014 09:30 25.06.2014 15:00	Running Abakan programming contest (interesting) Coming soon ACM ICPC World Finals 2014 (must coding!)
14.04.2013 13:13 3 Soft Parad 26.04.2013 09:00 27.04.2013 15:00 ISIT SFU Personal Olympiad 14.04.2013 10:00 14.04.2013 14:00 IMCS SFU training 14.04.2013 10:00 14.04.2013 15:00	Running IMCS SFU training (interesting) Running ISIT SFU Personal Olympiad (boring) Coming soon Soft Parad (boring)
11.11.1111 11:11 5 IPSC 01.01.0000 00:00 11.11.1111 11:11 Russian Code Cup 12.12.9999 23:58 12.12.9999 23:59 Facebook hacker cup 11.11.1111 11:11 04.04.4040 04:04 TopCoder Open 04.03.1234 12:21 04.03.4321 21:12 Codeforces round 156 11.11.1111 11:10 05.05.5555 05:55	Running Codeforces round 156 (interesting) Running Facebook hacker cup (interesting) Coming soon TopCoder Open (boring) Coming soon Russian Code Cup (must coding!)

Задача D. Перепрошивка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1.5 секунд
Ограничение по памяти:	256 мегабайт

Поликарп на днях приобрёл себе домашнего робота-андроида. Купил он его на распродаже, причём одну из немногих моделей — со встроенной кофе-машиной. Но в первые же дни он начал замечать, что зарядки его андроида хватает лишь на сутки, хотя в документации сказано, что её должно хватать на неделю. Обратившись в техподдержку, Поликарп выяснил, что функциональность робота зависит от кода прошивки, который записан в его памяти. Код представляет из себя строку из букв латинского алфавита. Изучив документацию на официальном сайте разработчиков, Поликарп нашёл код прошивки, который позволяет разблокировать такие функции, как пылесос, огнетушитель, которых как раз не хватает Поликарпу, а также продлить время работы до трёх дней, что вполне устроит Поликарпа.

Изучая документацию дальше, Поликарп узнал, что любого робота-андроида можно перепрошить. Перепрошивка представляет из себя сложный процесс и влечёт за собой изменение кода, а точнее переворот какой-то его подстроки (последовательности подряд идущих символов). Но там же сказано, что любой робот может выдержать не более четырёх перепрошивок, иначе он навсегда выйдет из строя.

Поликарп боится потерять своего робота, по-этому просит у Вас помощи. По имеющемуся коду и коду, который Поликарп хочет получить, выясните, можно ли превратить его робота в желаемого андроида не более чем за четыре перепрошивки.

Формат входных данных

В первой строке ввода содержится строка S — изначальный код прошивки андроида, во второй строке содержится строка T — код, о котором мечтает Поликарп. Каждая строка не пуста, имеет длину не более 32 символов и состоит из букв латинского алфавита.

Формат выходных данных

Выведите «Yes», если из S можно получить T , и «No» иначе.

Примеры

стандартный ввод	стандартный вывод
ABC abc	No
abcde deabc	Yes
abcd dbca	Yes
qwerty qwerty	Yes

Замечание

В первом примере невозможно получить желаемый код.

Во втором примере последовательность изменений такая: abcde, abced, cbaed, deabc.

В третьем: abcd, acbd, dbca.

В четвёртом примере код уже идеален.

Задача Е. Фонарики

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

Поликарпа недавно приняли на работу в одну престижную компанию. На собеседовании ему дали интересное задание:

Даны N лампочек, выстроенных в ряд. Изначально все они выключены. Далее происходит инверсия каждой первой, потом каждой второй, потом каждой третьей лампочки и так далее до каждой N -й. Требуется узнать сколько лампочек будет гореть после сделанных инверсий. Под инверсией понимается изменение состояния с «включено» на «выключено» и наоборот. Поликарп не задумываясь выдал ответ, и его приняли на место главного разработчика модуля, отвечающего за конфигурацию более серьёзных объектов — фонарей.

Скоро день города, поэтому фонари на главном авеню должны гореть не как обычно. Модуль, над которым работает Поликарп, должен принимать запросы вида i, j, k , запрос такого вида обозначает, что нужно применить инверсию к каждому k -му фонарю на отрезке $[i, j]$. Более точно, произойдёт инверсия i -го, $i + k$ -го, $i + 2k$ -го фонаря и так далее. Так как ответственный за подачу команд должен владеть информацией обо всех фонарях, модуль так же должен уметь выводить текущую конфигурацию всех фонарей.

Поликарп считает, что код его модуля идеален, но руководство считает, что он слишком медленно работает, и угрожает Поликарпу должностью полировщика нитей накала. Помогите Поликарпу улучшить его модуль, он в долгу не останется!

Формат входных данных

В первой строке ввода содержится натуральное число N — количество фонарей ($1 \leq N \leq 10^5$).

Во второй строке содержится начальная конфигурация фонарей в виде строки длиной N из символов «0» и «1», где «1» обозначает, что фонарь включён и «0», что выключен.

В третьей строке содержится целое число M — количество запросов ($0 \leq M \leq 10^5$). Далее в M строках идут запросы. Запрос на вывод обозначается словом `out`, запрос на изменение состоит из трёх целых чисел i, j, k ($1 \leq i \leq j \leq N, 1 \leq k \leq N$).

Формат выходных данных

На каждый запрос на вывод текущей конфигурации, выведите строку из символов «0» и «1», аналогично формату во входных данных.

Гарантируется, что потребуется вывести не более 2×10^5 символов.

Примеры

стандартный ввод	стандартный вывод
3 101 2 1 3 1 out	010
5 00000 4 2 4 2 out 1 5 2 out	01010 11111

Задача F. Задача без Поликарпа

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

Для заданного числа A найти такие N чисел x_i , что $x_1 \vee x_2 \vee \dots \vee x_N = A$ и $A > x_1 > x_2 > \dots > x_{N-1} > x_N \geq 0$.

Если таких вариантов несколько, найдите такой, у которого максимальное x_1 , если и таких вариантов больше одного, то тот, у которого максимальное x_2 , потом максимальное x_3 и так далее, то есть найдите лексикографически максимальный вариант.

Формат входных данных

В первой строке ввода через пробел содержатся два целых числа A ($1 \leq A \leq 9 \times 10^{18}$) и N ($2 \leq N \leq 10^4$).

Формат выходных данных

Выведите «-1» (без кавычек), если ответа не существует. Иначе выведите все x_i по порядку в одной строке через пробел.

Примеры

стандартный ввод	стандартный вывод
3 3	2 1 0
2 2	-1

Замечание

Результатом операции $a \vee b$ является побитовое «ИЛИ» чисел a и b . Например, $19_{10} \vee 5_{10} = 10011_2 \vee 101_2 = 10111_2 = 23_{10}$. В современных языках программирования эта операция обозначается как «or» или «|».