

Problem A. Идеальные наборы-2

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Искусство идеального взвешивания
по-прежнему является важнейшим из
всех искусств.
„История Новогиреевки“, том 2.*

На базарах города Новогиреевка все продавцы взвешивают товары с помощью двухчашечных весов и наборов гирь, каждая из которых весит целое число граммов. Суммарный вес всех гирь составляет N граммов. Такой набор называется *идеальным*, если любой груз весом в целое число граммов от 1 до N может быть уравновешен некоторым количеством гирь данного набора, и притом единственным образом. Гири разрешается класть на *обе* чашки весов, но гири одного веса на разные чашки ставить нельзя. Два способа взвешивания, отличающиеся лишь заменой некоторых гирь на другие того же веса, считаются одинаковыми. Так, для $N = 4$ идеальных наборов два: (1, 1, 1, 1) и (1, 3).

Найдите все идеальные наборы общим весом в N граммов, имеющие *наименьшее* количество гирь.

Input

В единственной строке записано одно целое число N — общий вес всех гирь в наборе ($1 \leq N \leq 10^5$).

Output

В первой строке запишите два числа: M — количество идеальных наборов, имеющих наименьшее количество гирь, и K — количество гирь в каждом таком наборе, затем в следующих M строках — идеальные наборы из K гирь общим весом N граммов, перечисленных в порядке неубывания веса. Все наборы выведите в лексикографическом порядке.

Examples

standard input	standard output
4	1 2 1 3
7	2 3 1 1 5 1 3 3

Problem B. Прочные штабеля

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

Способы формирования штабелей должны гарантировать безопасность работ, обеспечивать устойчивость штабелей и сохранность грузов. Фактические нагрузки от штабелей грузов на покрытия складов и площадок не должны превышать допустимых.

(Из «Инструкции по типовым способам и приемам погрузочно-разгрузочных работ при складировании грузов».)

В соответствии с технологической документацией каждый штабель формируется в точности из K коробок путем установки их друг на друга. Каждая коробка весит целое положительное число килограммов, вес штабеля должен составлять N килограммов. В целях обеспечения прочности и во избежание эффекта «расплющивания» вес каждой коробки должен быть не меньше суммарного веса всех находящихся над ней коробок. Например, для $N = 7$ существуют два прочных штабеля из $K = 3$ коробок: $7 = 1 + 2 + 4$ и $7 = 1 + 1 + 5$. (В каждом равенстве первое слагаемое — это вес верхней коробки, второе — вес второй сверху коробки, и, наконец, третье слагаемое указывает вес нижней коробки).

Всем штатным программистам необходимо подсчитать общее количество способов формирования указанных штабелей и довести до сведения главного технолога.

Input

В единственной строке записано два целых числа: N — вес штабеля и K — количество коробок в штабеле ($1 \leq N \leq 10^6, 1 \leq K \leq 20$).

Output

В единственной строке запишите количество способов формирования прочного штабеля по модулю $(10^9 + 9)$.

Examples

standard input	standard output
1 1	1
1 2	0
7 3	2

Problem C. Супердвоичная система счисления

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Все динозавры делятся на 10 групп — те, кто знают двоичную систему счисления, и те, кто не знают.
(Из «Большой энциклопедии динозавров».)*

Недавно палеонтологи обнаружили останки динозавра *Linhenykus monodactylus*, у которого на каждой передней конечности было только по одному пальцу. Распространение десятичной системы счисления связывают с количеством пальцев рук у человека. Значит, динозавры пользовались двоичной системой счисления. Точнее, супердвоичной системой, в которой для записи чисел использовались только «цифры» -1 , 0 или 1 . Супердвоичной записью числа N динозавры называли представление N в виде $2^k a_k + \dots + 2^2 a_2 + 2a_1 + a_0$, где каждое из чисел a_i равно -1 , 0 или 1 и $a_i \cdot a_{i+1} = 0$ для всех $0 \leq i \leq k-1$. Например, число 3 в этой системе записывалось в виде $10-1$, так как $3 = 2^2 \cdot 1 + 2 \cdot 0 + (-1)$.

Ваша задача — научиться записывать числа в супердвоичной системе динозавров.

Input

В единственной строке записано целое число N ($1 \leq N \leq 10^{18}$).

Output

Единственная строка содержит последовательность из разделенных пробелом целых чисел a_k, \dots, a_1, a_0 , образующих запись числа N в супердвоичной системе счисления. Число a_k является первой (слева) ненулевой цифрой в записи числа N , а a_0 — его последней цифрой (той, которая умножается на 1). Если таких представлений несколько, выведите любое из них.

Examples

standard input	standard output
1	1
3	1 0 -1

Problem D. Универсальное число (Division 1 Only!)

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Я придумал число-обезьянку
И число под названием дом.
И любую аптечную склянку
Обозначить хотел бы числом.*
(Н.Олейников.)

В итальянском городе Микель Анджело жил знаменитый скульптор. Отсекая лишнее, он создавал совершенное. Всё совершенное на свете измеряется универсальными числами, — говорил он. *Универсальным* он называл наименьшее натуральное число $U(N)$, из которого вычёркиванием цифр можно получить любое натуральное число от 1 до N . Порядок оставшихся цифр изменять нельзя. Например, для $N = 10$ универсальным будет число $U(10) = 1\,023\,456\,789$.

Ваша задача — научиться вычислять универсальные числа для любого натурального N .

Input

В единственной строке записано целое положительное число N , десятичная запись которого содержит не более 10^5 цифр.

Output

В единственной строке запишите универсальное число $U(N)$.

Examples

standard input	standard output
1	1
2	12
10	1023456789

Problem E. Носки

Input file: *standard input*
Output file: *standard output*
Time limit: 3 секунды
Memory limit: 256 мегабайт

Элитный бутик торгует носками одной модели разных размеров. Все они уложены на витрине в ряд. По новым правилам выкладки товара необходимо упорядочить все носки по неубыванию размера. Но всем прекрасно известно, что вложить носки в середину витрины невозможно, поэтому единственная доступная операция — вытащить любую пару носков и переложить её либо в начало витрины, либо в конец. Помогите мерчендайзеру бутика разложить носки по новым правилам за минимальное количество перемещений.

Input

В первой строке записано единственное целое число N ($1 \leq N \leq 2 \cdot 10^5$) — количество пар носков на витрине бутика. Во второй строке N целых чисел ($1 \leq A_i \leq 10^9$) — размеры пар носков в порядке их расположения на витрине.

Output

Единственное число — минимальное возможное количество перемещений носков.

Examples

standard input	standard output
3 1 3 2	1
4 2 1 3 2	2
2 2 1	1

Problem F. Большой тест

Input file: *standard input*
Output file: *standard output*
Time limit: 5 секунд
Memory limit: 256 мегабайт

На панели управления атомной электростанцией присутствуют N выключателей, каждый из которых может находиться в положениях «Включено» или «Выключено». После переключения i -го из них станция переходит в новый режим работы, который определяется номером i и состояниями всех переключателей.

Команда тестирования регулярно проверяет работу автоматики. Для этого тестировщикам необходимо проверить работу станции во всех возможных режимах. Поскольку тестирование проводится в пятницу вечером, необходимо провести всю операцию, используя минимально возможное количество переключений.

Input

В единственной строке — целое число N ($1 \leq N \leq 18$).

Output

В первой строке целое число M — минимальное требуемое количество переключений. В следующей строке M целых чисел a_i ($1 \leq a_i \leq N$) — номера выключателей.

Examples

standard input	standard output
1	2 1 1
2	8 2 1 2 2 1 2 1 1

Problem G. Доброе утро, коллега!

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

В головном офисе ICL столы расположены таким образом, что к любому рабочему месту от входной двери можно подойти единственным способом. Офис можно представить в виде дерева, в некоторых вершинах которого расположены рабочие места, а длина каждого ребра равна 1 метру. Когда сотрудники утром приходят на работу, каждый из них считает своим долгом подойти ко всем, кто пришел раньше него, чтобы лично поприветствовать и пожелать хорошего дня, и лишь затем пройти на своё место.

Однако, администратору офиса не очень нравится эта традиция. Он считает, что сотрудники тратят слишком много рабочего времени на личные приветствия, и решил подготовить служебную записку директору компании. В приложении 15 к служебной записке администратор собирается привести информацию о том, какое расстояние проходит утром каждый сотрудник от входной двери до своего рабочего места. К счастью, последовательность прихода сотрудников уже известна, как и тот факт, что каждый из них выбирает для утреннего приветствия кратчайший путь и успевает занять своё место в офисе до прихода следующего.

Input

В первой строке целое число N ($2 \leq N \leq 10^5$) — количество вершин в дереве, описывающем офис. Далее в каждой из $N - 1$ строк два различных целых числа a и b ($1 \leq a, b \leq N$) — номера вершин, соединенных ребром. В следующей строке целое число M ($1 \leq M \leq N - 1$) — количество сотрудников, работающих в офисе. В следующей строке M различных целых чисел v_i ($2 \leq v_i \leq N$) — номера вершин, где расположены рабочие места сотрудников. Они перечислены в порядке прихода сотрудников в офис. (Вершина 1 — входная дверь, все входят через нее.)

Output

В единственной строке M целых чисел: расстояния в метрах, пройденные каждым сотрудником, в порядке их нумерации.

Examples

standard input	standard output
7 1 2 1 3 3 4 3 5 3 6 6 7 6 6 4 2 7 3 5	2 4 7 7 9 10

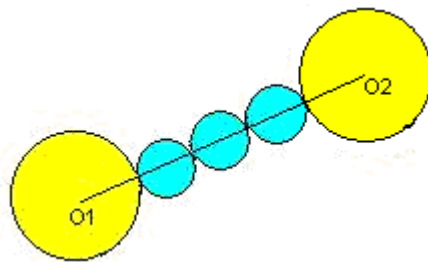
Problem H. Крепость (Division 1 Only!)

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 256 мегабайт

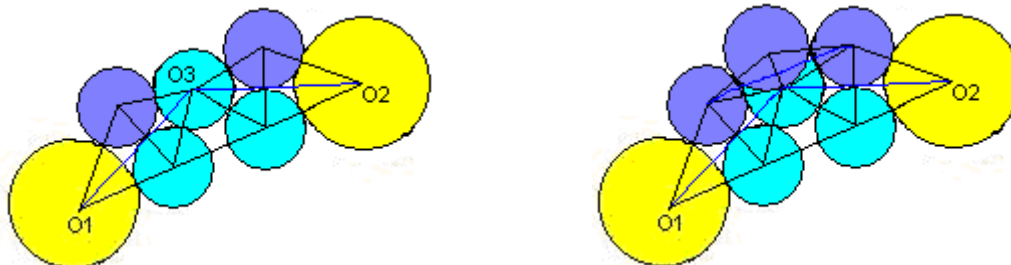
В средние века на Руси крепости окружались забором из бревен. Иногда крепости осаждали супостаты и сжигали их. Тогда забор из бревен выгорал, но самые толстые бревна оставались на месте. Одну из таких заброшенных крепостей решили отстроить заново и начали с забора из бревен.

Вид крепости сверху представляет собой несколько кругов, сохранившихся от старого частокола бревен. Последовательно соединим отрезками центры кругов в том порядке, в каком они стояли в старом частоколе, и получим выпуклый многоугольник. На каждом отрезке, соединяющем центры двух соседних сохранившихся столбов, поступаем следующим образом: бревна приставляем вплотную друг к другу и к существующим столбам вдоль отрезка до тех пор, пока возможно это делать, не нарушая симметрию относительно серединного перпендикуляра к отрезку. При этом центры новых кругов должны лежать строго на отрезке.

Если центры всех кругов лежали на отрезке и круги касались друг друга, то считалось, что строителям крупно повезло, и можно было бодро переходить к следующему проему.



Если же в середине оставался просвет, то его заполняли симметрично относительно того же серединного перпендикуляра, приставив одно или пару бревен с наружной стороны крепости. Если соединить последовательно центры всех кругов строящегося проема, то получается ломаная. Необходимо было сделать ее выпуклой. Для этого вогнутый внутрь крепости участок ломаной, состоящий из двух звеньев, симметрично отображали относительно прямой, проходящей через крайние вершины этого участка. Процесс симметричного отображения продолжали до тех пор, пока не получалась выпуклая ломаная. При таких действиях длина ломаной не менялась.



Чтобы покрыть бревна снаружи специальным негорючим составом, необходимо подсчитать внешний периметр стен восстановленной крепости. Гарантируется, что на итоговом чертеже не имеется наложений кругов.

Input

В первой строке единственное целое число N ($1 \leq N \leq 100$) — количество сохранившихся бревен.

В следующих N строках по два целых числа X_i, Y_i ($0 \leq X_i, Y_i \leq 1000$) — координаты центров сохранившихся бревен в порядке обхода по часовой стрелке.

В последней строке два целых числа: R_0 ($1 \leq R_0 \leq 10$) — радиус заготовленных бревен и R ($R_0 \leq R \leq 10$) — радиус сохранившихся бревен.

Output

Выведите минимальное количество бревен, необходимых для постройки забора вокруг крепости, и длину внешнего периметра с точностью до 10^{-6} .

Examples

standard input	standard output
4 0 0 0 6 6 6 6 0 1 2	4 50.265482
4 0 0 0 6 6 6 6 0 1 2	4 50.265482

Problem I. Бактерии

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 256 мебибайт

Знаменитый биолог доктор Скальпель изучал бактерию, поднятую со дна Марианской впадины. Странная оказалась бактерия. Она всегда находилась в одном из двух состояний: «покой» или «возбуждение».

Если бактерия была в состоянии возбуждения, то посылала электромагнитный импульс соседним бактериям.

Если какая-либо бактерия получала импульсы от двух соседних бактерий, то в следующую секунду бактерия переходила в состояние покоя, а если от одной соседней, то бактерия переходила в состояние возбуждения независимо от того состояния, в котором она сама находилась. Бактерия, не получившая ни одного импульса, в следующую секунду будет покоем.

Доктор Скальпель решил поставить эксперимент. Он расположил в ряд N бактерий, некоторые из которых находились в состоянии возбуждения, и начал наблюдать.

Каждую секунду он записывал в тетрадочку расположение бактерий, находящихся в состоянии возбуждения. Сделать это было довольно легко, так как возбужденные бактерии светились приятным для глаза зеленым цветом.

Доктору хочется узнать состояние бактерий через T секунд от начала эксперимента. Но у него нет времени ждать. Помогите ему.

Input

В первой строке два целых числа T и N ($0 \leq T \leq 10^{18}$, $1 \leq N \leq 10^5$) — количество прошедших секунд и количество бактерий в ряду. Вторая строка состоит из N символов «0» или «1» и описывает начальное состояние бактерий («1» — бактерия в состоянии возбуждения, «0» — в состоянии покоя).

Output

Состояние бактерий через T секунд после начала эксперимента в формате, аналогичном формату входных данных.

Examples

standard input	standard output
2 5 00100	10001
0 1 1	1
1 5 01110	11011

Problem J. БДП

Input file: *standard input*
Output file: *standard output*
Time limit: 3 секунды
Memory limit: 256 мегабайт

Бинарное дерево поиска — это двоичное дерево, для которого выполняются следующие дополнительные условия:

1. Оба поддерева — левое и правое — являются бинарными деревьями поиска.
2. У всех узлов левого поддерева произвольного узла X значения ключей данных меньше, чем значение ключа данных самого узла X .
3. У всех узлов правого поддерева того же узла X значения ключей данных больше, чем значение ключа данных узла X .

В языке деревостроения описаны несколько команд:

- **ADD X** — находит в текущем дереве поиска место для ключа X и записывает его в новом узле. Если в дереве нет ни одного узла, то создается дерево из единственного узла с ключом, равным X . Если ключ X уже присутствует в дереве, то команда возвращает **FALSE**. В остальных случаях команда возвращает **TRUE**.
- **FIND X** — ищет в текущем дереве ключ X и возвращает **TRUE**, если он найден, и **FALSE** в противном случае.
- **REMOVE X** — ищет в текущем дереве ключ X . Возвращает **TRUE** или **FALSE** аналогично команде **FIND**, после чего удаляет узел с ключом X . Удаление узла происходит следующим образом. Если узел не имеет потомков, то узел перестает существовать и операция заканчивается. Если у узла есть только один потомок, то поддерево этого потомка переносится на место удаляемого узла и операция завершается. Если у узла два потомка, то на место удаленного ключа записывается ключ из самого левого узла правого поддерева (наименьшее число правого поддерева), после чего производится операция удаления самого левого узла правого поддерева.

Высота корня дерева равна единице, а высота произвольного узла на единицу меньше высоты корня любого из его поддеревьев (левого и правого).

Назовем стоимостью каждой команды максимальную высоту узла, который обрабатывается при выполнении команды:

- Для команды **ADD** — высоту узла с ключом X после выполнения команды.
- Для команды **FIND** — высоту узла с ключом X , либо высоту узла, отсутствие потомка у которого свидетельствует об отсутствии ключа X в дереве поиска. Стоимость команды на пустом дереве равна единице.
- Для команды **REMOVE** — высоту узла, который будет удален, либо высоту узла, отсутствие потомка у которого свидетельствует об отсутствии ключа X в дереве поиска. Стоимость команды на пустом дереве равна единице.

Задача — обработать все команды во входном файле и для каждой вывести ответ и стоимость. Изначально в дереве нет ничего.

Input

В первой строке единственное целое число N ($1 \leq N \leq 10^5$) — количество команд. Далее N строк, содержащих одну из трех команд {ADD, FIND, REMOVE} с отделенным пробелом целочисленным аргументом X ($-10^9 \leq X \leq 10^9$).

Output

Для каждой команды входного файла вывести в отдельной строке ответ (TRUE или FALSE) и стоимость, разделенные пробелом.

Examples

standard input	standard output
3 ADD 6 FIND 6 REMOVE 6	TRUE 1 TRUE 1 TRUE 1
11 REMOVE 7 FIND 8 ADD 9 ADD 10 ADD 8 FIND 9 FIND 8 FIND 10 FIND 11 REMOVE 9 FIND 8	FALSE 1 FALSE 1 TRUE 1 TRUE 2 TRUE 2 TRUE 1 TRUE 2 TRUE 2 FALSE 2 TRUE 2 TRUE 2

Problem K. Дорога приключений

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

*Когда-то и меня вела дорога приключений, а
потом мне прострелили колено.*

Любимая фраза стражников Скайрима.

*В поле бес нас водит, видно,
Да кружит по сторонам.*

Пушкин А.С.

Мы вышли из леса и остановились в точке A на дороге № 1. Искатели приключений! Нам хотелось как можно быстрее побывать на всех незнакомых дорогах. Не раздумывая, из точки A мы двинулись в сторону дороги № 2, в строго перпендикулярном направлении к ней. Жажда приключений заставляла двигаться дальше и дальше. Добравшись до дороги № 2, мы отправились к дороге № 3 в перпендикулярном к ней направлении... Мы прошли сотню-другую этих бесконечных, похожих друг на друга, прямолинейных дорог... Оказавшись на последней из них с номером n , мы аналогичным способом отправились на дорогу № 1. И тут нас ждал сюрприз! Неожиданно для себя мы оказались в начальной точке A нашего путешествия. Размышляя над этим удивительным фактом, мы смогли точно определить координаты точки A на дороге № 1.

Ваша задача — вычислить координаты этой замечательной точки.

Input

В первой строке содержится одно натуральное число n — количество дорог, которые нам предстояло пройти ($2 \leq n \leq 10\,000$). В каждой i -ой из n следующих строк записаны координаты X_{i1} , Y_{i1} и X_{i2} , Y_{i2} двух различных точек, через которые проходит i -ая дорога. Координаты всех точек — целые числа, не превосходящие по модулю 100. Все числа разделены пробелом. Гарантируется, что хотя бы две дороги не параллельны и не совпадают.

Output

Выведите через пробел координаты точки A , расположенной на первой дороге. Ответ считается верным, если абсолютная или относительная погрешность не превосходит 10^{-5} . Если возможных решений несколько, выведите любое из них. Если решения не существует, выведите -1 .

Examples

standard input	standard output
3 -1 0 1 0 1 0 0 1 0 1 -1 0	0.00000 0.00000
2 0 0 2 0 1 1 0 2	2.00000 0.00000

Problem L. ШахМады

Input file: *standard input*
Output file: *standard output*
Time limit: 1 секунда
Memory limit: 256 мегабайт

На шахматной доске размера $N \times N$ клеток расставляются шахматные фигуры. В игре принимает участие единственный игрок, которому доступны 4 вида фигур одного цвета: пешка, конь, ладья и король. В любой момент времени на каждой клетке поля может находиться одна из фигур, либо клетка может оставаться свободной.

К сожалению, описание правил расстановки и передвижения фигур в этой игре запрещено Роскомнадзором на территории Российской Федерации. Нам доступна лишь последняя глава из правил — «Оценка результатов игры и подсчёт очков». Она гласит:

«Для подсчёта результатов игры подсчитайте суммарную боевую мощь всех фигур, находящихся на поле. Каждая фигура, оставшаяся в конце игры на поле, прибавляет к боевой мощи определённое количество очков. Так, пешка прибавляет одно очко к боевой мощи, конь прибавляет два очка, ладья — три, король является самой сильной фигурой и его боевая мощь равна четырём. В том случае, если количество фигур на поле больше или равно количеству пустых клеток, игрок признаётся победителем в партии и ему присваивается столько очков, сколько боевой силы дают все его фигуры, находящиеся на поле. Если же пустых клеток на поле оказалось больше, чем фигур, то игроку присуждается поражение.»

Ваша задача — по текущему состоянию игрового поля определить результат игры, если она закончится прямо сейчас.

Input

В первой строке записано единственное целое положительное число $N < 14$. В последующих N строках записано по N чисел, разделенных пробелом, задающих расстановку фигур на поле, где 0 (ноль) означает, что поле пусто, 1 соответствует пешке, 2 — коню, 3 — ладье и 4 указывает на наличие в данном поле короля.

Output

Выведите единственное число — количество очков, заработанных игроком в партии. В случае поражения игрока выведите -1.

Examples

standard input	standard output
7 0 0 0 1 0 0 0 0 4 0 1 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 1 0 3 0 0 0 0 1 0 0 0	-1
5 0 2 4 2 3 1 0 1 0 1 0 0 0 0 0 1 1 0 1 1 3 2 4 0 3	30