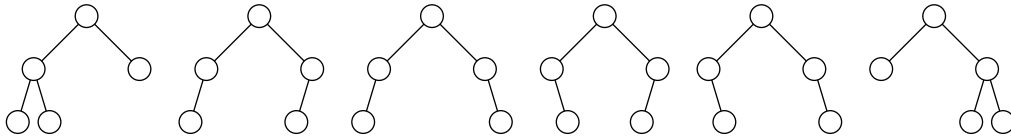


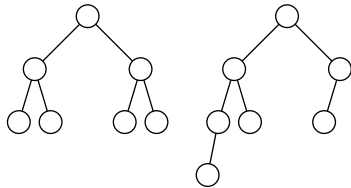
Задача А. AVL

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

AVL деревья, придуманные российскими учёными Адельсон-Вельским и Ландисом, являются примером сбалансированного бинарного дерева поиска. В терминологии AVL, подвешенное бинарное дерево называется *сбалансированным*, если для каждой вершины высоты её левого и правого поддеревьев отличаются не более, чем на один. Такое дерево, собственно, и называется *AVL-деревом*. Разумеется, существует далеко не единственное AVL-дерево при фиксированном числе вершин. К примеру, существует шесть AVL-деревьев с пятью вершинами, они изображены на рисунке ниже.



Деревья с одинаковым числом вершин могут иметь разную высоту, к примеру, на рисунке снизу нарисовано два дерева с семью вершинами, которые имеют высоты 2 и 3, соответственно.



Вам даны два числа — N и H , требуется найти число AVL-деревьев, которые состоят из N вершин и имеют высоту H . Поскольку их число довольно велико, выведите искомое количество по модулю 786 433.

Формат входного файла

Единственная строка входного файла содержит два числа — N и H ($1 \leq N \leq 65\,535$, $0 \leq H \leq 15$).

Формат выходного файла

Выведите единственное число — количество AVL деревьев с N вершинами высоты H , по модулю 786 433.

Примеры

| avl.in | avl.out |
|--------|---------|
| 7 3 | 16 |

Note

786 433 простое число, и $786\,433 = 3 \cdot 2^{18} + 1$.

Задача В. Простые сложности

Имя входного файла: again.in
Имя выходного файла: again.out
Ограничение по времени: 5 секунды
Ограничение по памяти: 256 мегабайта

В этой жизни не всё так просто. Особенно числа. Вам дан набор чисел. Необходимо для каждого из них определить, является ли оно простым.

Формат входного файла

В первой строке входных данных содержится единственное число $1 \leq T \leq 5\,000$ — количество чисел, которые необходимо проверить на простоту. Далее содержится T целых положительных чисел, не превосходящих 10^{18} .

Формат выходного файла

В i -й строке выходных данных должно быть записано «YES», если i -е число является простым, и «NO» в противном случае.

Примеры

| again.in | again.out |
|----------|-----------|
| 2 | YES |
| 3 | NO |
| 4 | |

Задача С. ДНК Роботов

Имя входного файла: robots.in
Имя выходного файла: robots.out
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Последние достижения в технологии синтеза ДНК позволили провести эксперимент по созданию биороботов.

Для облегчения задачи создания ПО для управления роботами было принято решение, что их ДНК будет состоять из $M = 2^n$ символов для некоторого $n \geq 2$. Кроме этого, по техническим причинам это будет не обычная строка, а циклическая, то есть её можно начинать читать с любой позиции.

Одной из целей эксперимента является изучение мутаций биороботов. В результате продолжительных наблюдений было найдено много различных видов роботов. Для понимания

процесса мутации учёным необходимо решить следующую задачу. Для ДНК двух роботов требуется определить коэффициент их схожести. Он вычисляется, как максимальное количество совпадающих символов при наилучшем совмещении этих ДНК. Чем больше символов совпадает, тем лучше совмещение.

Требуется написать программу, которая найдёт наилучшее совмещение двух ДНК.

Формат входного файла

В первой строке входного файла записано одно число M ($4 \leq M \leq 131072$). В следующих двух строках записаны ДНК двух роботов. Обе ДНК — строки, состоящие ровно из M символов из множества {'A', 'C', 'G', 'T'}.

Формат выходного файла

В выходной файл выведите два числа — максимальное количество совпадающих символов и значение оптимального сдвига — неотрицательное количество символов второй ДНК, которые необходимо перенести из конца строки в её начало для достижения наилучшего совмещения.

Примеры

| robots.in | robots.out |
|--|------------|
| 16 ACGTACGTACGTACGT CGTACGTACGTACGTC | 15 1 |

Задача D. Задача для второклассника

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

Вам даны два числа. Необходимо найти их произведение.

Формат входного файла

Входные данные состоят из двух строк, на каждой из которых находится целое одно целое число, длина которого не превосходит двухсот пятидесяти тысяч символов.

Формат выходного файла

Выведите произведение данных чисел.

Примеры

| multiply.in | multiply.out |
|-------------|--------------|
| 2 2 | 4 |

Задача E. Факторизация

Имя входного файла: pollard.in
Имя выходного файла: pollard.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дано натуральное число. Факторизуйте его, то есть представьте в виде произведения набора простых чисел. Число p называется простым, если имеет ровно два различных натуральных делителя: 1 и p .

Формат входного файла

В единственной строке записано единственное натуральное число N . $2 \leq N \leq 9 \cdot 10^{18}$.

Формат выходного файла

Выведите в неубывающем порядке одно или несколько простых чисел, произведение которых равно N .

Примеры

| pollard.in | pollard.out |
|------------|-------------|
| 6 | 2 3 |
| 7 | 7 |

Задача F. Дискретное логарифмирование

Имя входного файла: log.in
Имя выходного файла: log.out
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Даны натуральные числа a , b , n . Требуется найти *дискретный логарифм* b по основанию a по модулю n , то есть такое число x ($0 \leq x < n$), что $a^x \equiv b \pmod{n}$.

Формат входного файла

В первой строке входного файла заданы через пробел три целых числа a , b и n ($0 \leq a, b, n \leq 10^{12}$), $n \neq 0$.

Формат выходного файла

В первой строке выходного файла выведите -1 , если дискретного логарифма не существует. Иначе следует вывести его значение.

Если ответ неоднозначен, разрешается выводить любой.

Примеры

| log.in | log.out |
|--------|---------|
| 2 4 6 | 2 |

Задача G. Задача для шестиклассника

Имя входного файла: `sqrt.in`
Имя выходного файла: `sqrt.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Вам требуется найти такое x , что:

$$x^2 = a \pmod{m}$$

Формат входного файла

В первой строке входного файла записано число k ($1 \leq k \leq 50$) — количество тестовых наборов во входном файле. Далее идут k строк, описывающих тестовые наборы. Каждый набор содержит 2 числа — a и m ($0 \leq a \leq 10^9$, $1 \leq m \leq 10^9$).

Формат выходного файла

Для каждого набора выведите число x или IMPOSSIBLE, если такого x не существует.

Примеры

| sqrt.in | sqrt.out |
|---------|------------|
| 3 | 1 |
| 1 3 | 4 |
| 7 9 | IMPOSSIBLE |
| 2 4 | |

Задача H. Teams

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

There are $3 \cdot N$ same year students studying in two specializations — mathematics (M) and programming (P). Let's call students of the first specialization mathematicians and students of the second specialization programmers. There are several groups of each specialization. The number of groups is the same for both specializations. The number of students in each group is greater than or equal to three.

The students are loaded much by the studies, that's why they can meet each other only during classes. They have separate classes for each group except physical education (P. E.). P. E. classes are held for pairs of groups of different specialization and each group share its P. E. classes with only one other group. All groups attend P. E. classes.

All the students are very responsible and attend all the classes. When two students meet each other at the same class, they become familiar.

There is a head in each group. All the group heads attend head meetings, where all of them meet each other and become familiar.

You should form N teams consisting of three students each so that each student is used in exactly one group. There should be at least one mathematician and at least one programmer in each team. All the members of one team should be familiar to each other.

Формат входного файла

The first line contains 2 integers K and M ($1 \leq K, M \leq 1000$, $K = 3 \cdot N$) — the total number of students and the number of pairs of familiar students correspondingly. The second line contains K characters 'M' or 'P', i -th character describes i -th student's specialization. The following M lines define pairs of familiar students. It's guaranteed that the input describes student's specializations and relationships that do not contradict the problem statement.

Формат выходного файла

The first line should contain "Yes" or "No" depending on the possibility to form N teams. If such possibility exists, the following N lines should contain numbers of the students of each team, three numbers in each line. If there are several solutions, output any.

Примеры

| teams.in | teams.out |
|--------------|-----------|
| 12 34 | Yes |
| ММРРРММММРРР | 4 5 1 |
| 1 2 | 6 2 3 |
| 1 3 | 10 12 7 |
| 1 4 | 9 8 11 |
| 1 5 | |
| 1 6 | |
| 1 7 | |
| 1 10 | |
| 2 3 | |
| 2 4 | |
| 2 5 | |
| 2 6 | |
| 3 4 | |
| 3 5 | |
| 3 6 | |
| 4 5 | |
| 4 6 | |
| 4 7 | |
| 4 10 | |
| 5 6 | |
| 7 8 | |
| 7 9 | |
| 7 10 | |
| 7 11 | |
| 7 12 | |
| 8 9 | |
| 8 10 | |
| 8 11 | |
| 8 12 | |
| 9 10 | |
| 9 11 | |
| 9 12 | |
| 10 11 | |
| 10 12 | |
| 11 12 | |