Задача A. Треугольник

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

Дан строго выпуклый n-угольник. Найдите площадь максимального треугольника, лежащего внутри (или на границе) этого n-угольника.

Формат входных данных

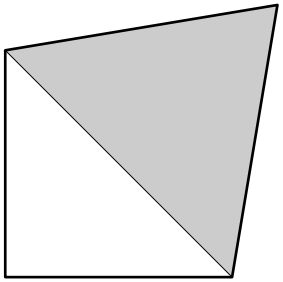
В первой строке находится натуральное число n — количество вершин ($3 \leqslant n \leqslant 2\,000$). Далее заданы координаты вершин в порядке обхода по или против часовой стрелки: по два целых числа, не превышающих по модулю 10^7, на вершину.

Формат выходных данных

Выведите одно вещественное число — искомую максимальную площадь. Ваш ответ будет считаться верным, если он отличается от правильного не более, чем на 10^{-3}.

Пример

стандартный ввод	стандартный вывод
4 0 0 10 0 12 12 0 10	70.0



Задача B. Палиндромы Фибоначчи

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

Рассмотрим следующие строки, состоящие из цифр 0 и 1:

$S_0 =$ “0”

$S_1 =$ “01”

$S_2 = S_1 S_0 =$ “010”

$S_3 = S_2 S_1 =$ “01001”

$S_4 = S_3 S_2 =$ “01001010”

$S_5 = S_4 S_3 =$ “0100101001001”

...

$S_n = S_{n-1} S_{n-2}$

...

Как видно из определения, строка S_{n-1} является префиксом строки S_n для любого $n > 0$. Значит, существует бесконечная строка S_∞, являющаяся *пределом* строк S_n, то есть такая, что для любого $n \geqslant 0$ строка S_n является префиксом S_∞.

(Если предыдущее определение не удалось понять, то можете считать, что S_∞ — это просто S_n для достаточно большого n, скажем, $n = 10^{100}$)

Строка S_∞ называется *строкой Фибоначчи*.

Вам даны несколько подстрок строки Фибоначчи. Выясните, какие из них являются палиндромами. Строка называется *палиндромом*, если её первый символ совпадает с последним, второй — с предпоследним, и так далее.

Формат входных данных

В первой строке находится натуральное число n — количество подстрок ($1 \leqslant n \leqslant 10^4$). Далее заданы сами строки. Каждая из них задана двумя числами *start* и *len* и соответствует подстроке строки S_∞, начинающейся с позиции *start* (позиции нумеруются с единицы) и состоящей из *len* символов ($1 \leqslant start, len \leqslant 10^9$).

Формат выходных данных

Выведите n строк, по одной для каждой входной подстроки. Для очередной подстроки выведите "TAK", если она является палиндромом, и "NIE" в противном случае.

Пример

стандартный ввод	стандартный вывод
4 2 4 3 5 3 7 1 11	TAK NIE TAK TAK

Задача C. Вариация Нима

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

На столе лежат n кучек камней: a_1 камней в первой кучке, a_2 камней во второй, ..., a_n в n-ой. Двое играют в игру, делая ходы по очереди. За один ход игрок может либо взять произвольное ненулевое количество камней (возможно, все) из одной любой кучки, либо произвольным образом разделить любую существующую кучку, в которой не меньше двух камней, на две непустые кучки. Проигрывает тот, кто не может сделать ход. Кто выигрывает при правильной игре?

Формат входных данных

В первой строке задано целое число t — количество тестов ($1 \leqslant t \leqslant 100$). Следующие t строк содержат сами тесты. Каждая из них начинается с целого числа n — количества кучек ($1 \leqslant n \leqslant 100$). Далее следует n целых чисел a_1, a_2, ..., a_n через пробел — количество камней в кучках ($1 \leqslant a_i \leqslant 10^9$).

Формат выходных данных

Выведите t строк; в i-ой строке выведите "FIRST", если в i-ом тесте при правильной игре выигрывает первый игрок, и "SECOND", если второй.

Пример

стандартный ввод	стандартный вывод
3 1 1 2 1 1 3 1 2 3	FIRST SECOND FIRST

Решение, работающее для всех тестов, в которых $1 \leqslant a_i \leqslant 1000$, получит не менее 60 баллов.

Задача D. Система контроля версий

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

Группа разработчиков решила использовать систему контроля версий. Для нужд задачи будем считать, что в системе контроля версий хранится массив. Вначале массив состоит из нулей. Версии массива будем обозначать строчками из строчных латинских букв и цифр. Вы должны поддерживать следующие операции:

1. Изменить *заданный* элемент массива *заданной* версии на *заданное* число и присвоить полученному массиву новую версию.
2. Сообщить *заданный* элемент массива *заданной* версии.

Отметим, что, пока вы не выполните очередную операцию, информация о следующей операции доступна не будет.

Заранее известно, что «последних» версий массива (то есть версий, из которых не было сделано ни одного изменения) в любой момент времени будет не более 20.

Формат входных данных

В первой строке задано натуральное число n — длина массива ($1 \leqslant n \leqslant 100\,000$). Во второй строке задано название s начальной версии.

В третьей строке задано натуральное число k ($1 \leqslant k \leqslant 100\,000$) — количество операций.

После считывания этих данных вы должны организовать интерактивное общение с тестирующей системой. Вы должны k раз проделать следующую последовательность действий:

- считать описание очередной операции,
- выполнить эту операцию,
- если необходимо, вывести очередной ответ на отдельной строке,
- в случае вывода данных, выполнить команду `flush(output)` в Паскале и `fflush(stdout)` в C и C++.

Операции задаются в формате '`S v p q w`' (записать число q в ячейку p массива версии v и присвоить полученному массиву версию w) и '`G u p`' (вывести значение ячейки p массива версии u). Гарантируется, что все требуемые операции корректны: любое название в первый раз встречается во входных данных в качестве новой версии (обозначенной выше w) или в качестве начальной версии (обозначенной выше s), любая версия становится новой/начальной ровно один раз, $1 \leqslant p \leqslant n$.

Гарантируется, что после любой операции существует не более 20 версий, которые никогда не появлялись в качестве исходной версии (обозначенной выше v) для операций типа 'S'.

Не рекомендуется использовать потоки ввода/вывода C++ (`fstream`), так как они работают существенно медленней `FILE *` и могут стать причиной превышения ограничения по времени.

Названия версий непусты, состоят из цифр и строчных букв латинского алфавита, и их длины не превосходят 10 символов. Ячейки массива нумеруются с единицы. Числа, записываемые в ячейки, находятся в диапазоне от -2^{31} до $2^{31}-1$.

Формат выходных данных

Смотрите описание входного файла.

Пример

стандартный ввод	стандартный вывод
5 start 6 S start 2 10 v1 S start 2 15 v2 S v1 1 40 v3 G v3 2 G start 1 G v1 1	10 0 0