A. ACM ICPC

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

В одном маленьком, но очень гордом ВУЗе было принято решение выиграть ACM ICPC. Для этого нужно составить как можно больше команд из трёх человек, но так как желающих студентов всего 6, было решено составить две команды.

Участнику с номером i по результатам тестирования была присвоена некоторая *сила* a_i. *Силой команды* называется сумма сил всех участников в этой команде. Теперь руководству ВУЗа интересно, возможно ли собрать две команды с одинаковой силой. Ответьте на этот вопрос руководства.

Входные данные

В единственной строке находятся шесть целых чисел $a_1, ..., a_6$ $(0 \le a_i \le 1000)$ — силы участников.

Выходные данные

Выведите «YES» (без кавычек), если из данных участников можно собрать две команды с одинаковой силой, и «NO» (без кавычек) — иначе.

Вы можете выводить каждую букву в любом регистре (строчную или заглавную).

Примеры

входные данные	Скопировать
1 3 2 1 2 1	

выходные данные
YES

входные данные	Скопировать
1 1 1 1 1 99	

выходные данные
NO

Примечание

В первом тесте можно взять в первую команду 1-го, 2-го и 6-го участника, во вторую — 3-го, 4-го и 5-го: силы команд будут $1 + 3 + 1 = 2 + 1 + 2 = 5$.

Во втором тесте участник номер 6 слишком сильный и его команда будет заведомо сильнее другой.

B. Влад и столовые

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Влад очень любит ходить в столовые. За свою жизнь он сделал это n раз. К сожалению, в последнее время он начал замечать, что его выбор не отличается разнообразием. Ему хочется это исправить, и поэтому он провёл следующее исследование.

Сначала Влад присвоил каждой столовой индивидуальный номер. Затем он выписал в ряд все номера столовых, в которых он был, в порядке посещения. Теперь он хочет найти такую столовую, что последний раз он в ней был раньше последних посещений всех остальных столовых. Иными словами, он хочет найти столовую, в которой не был дольше всего на текущий момент. Помогите ему в этом.

Входные данные
В первой строке находится одно целое число n ($1 \le n \le 2 \cdot 10^5$) — количество выписанных Владом номеров столовых.

Во второй строке находятся n целых чисел $a_1, a_2, ..., a_n$ ($0 \le a_i \le 2 \cdot 10^5$) — номера столовых в порядке посещения Владом. Влад мог посетить одну и ту же столовую несколько раз. Обратите внимание, не обязательно, что столовые пронумерованы подряд.

Выходные данные
Выведите одно число — номер столовой, в которой Влад не был как можно дольше.

Примеры

входные данные	Скопировать
5 1 3 2 1 2	
выходные данные	
3	

входные данные	Скопировать
6 2 1 2 2 4 1	
выходные данные	
2	

Примечание
Рассмотрим первый тест. В нём есть три столовые, причём последнее посещение столовых с номерами 1 и 2 состоялись после последнего посещения столовой с номером 3, поэтому эта столовая — искомая.

Во втором тесте также есть три столовые с номерами 1, 2 и 4. Столовые с номерами 1 и 4 были посещены после последнего посещения столовой с номером 2, поэтому ответ 2. Обратите внимание, что Влад мог пропустить некоторые номера в нумерации столовых.

C. Петя и катакомбы

ограничение по времени на тест: 1 секунда
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Однажды Петя — очень храбрый исследователь — решил заняться изучением великих парижских катакомб. Так как Петя не очень опытный в этом деле, его исследование заключается в хождении по катакомбам.

Катакомбы представляют из себя набор комнат, некоторые из которых соединены переходами, переход может соединять комнату с самой собой, по переходам можно ходить в любую сторону. Переходы могут находиться на разных глубинах, что позволяет им не пересекаться. Каждую минуту Петя выбирает произвольным образом переход из комнаты, в которой он находится, и проходит его за одну минуту. Когда он заходит в комнату в минуту i, он делает в журнал запись — число t_i:

- Если Петя уже был в этой комнате, он записывает минуту, когда он был в этой комнате в предыдущий раз;
- Иначе Петя записывает в журнал произвольное неотрицательное целое число, строго меньшее текущей минуты i.

Изначально Петя находился в какой-то из комнат в минуту 0, при этом число t_0 он не записывал.

В какой-то момент Петя устал, бросил журнал и пошёл домой (не обязательно в той же комнате, из которой он начинал). Вася нашёл этот журнал, и теперь ему интересно: какое наименьшее количество комнат может быть в катакомбах, согласно журналу Пети?

Входные данные
В первой строке находится одно целое число n ($1 \le n \le 2 \cdot 10^5$) — количество записей в журнале Пети.

Во второй строке находятся n целых чисел $t_1, t_2, \ldots, t_n$ ($0 \le t_i < i$) — записи в журнале.

Выходные данные
В первой строке выведите единственное число — минимальное возможное количество комнат в парижских катакомбах.

Примеры

входные данные	Скопировать
2 0 0	

выходные данные
2

входные данные	Скопировать
5 0 1 0 1 3	

выходные данные
3

Примечание
В первом тестовом примере последовательность комнат, по которым прошёл Петя, могла быть $1 \to 1 \to 2$ или $1 \to 2 \to 1$, или, например, $1 \to 2 \to 3$. Минимальное число комнат равно 2.

Во втором тестовом примере эта последовательность могла быть $1 \to 2 \to 3 \to 1 \to 2 \to 1$.

D. Восстановление строки

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Назовем подстроку некоторой строки самой частой, если количество ее вхождений не меньше количества вхождений любой другой подстроки.

Дано множество строк. Назовем какую-то строку хорошей, если все элементы множества являются ее самыми частыми подстроками. Восстановите непустую минимальную по длине хорошую строку, а из таких — лексикографически минимальную. Если же таких строк не существует, то выведите «NO» (без кавычек).

Подстрокой называется непрерывная подпоследовательность букв строки. Например, «ab», «c», «abc» являются подстроками строки «abc», а «ac» — нет.

Количество вхождений подстроки в строку равно количеству таких позиций в данной строке, в которых встречается данная подстрока. Вхождения подстроки могут перекрываться.

Строка a лексикографически меньше строки b, если a является префиксом b, или в a стоит меньшая буква на первой позиции, где a и b отличаются.

Входные данные

В первой строке дано целое число n ($1 \le n \le 10^5$) — количество строк в данном множестве.

Далее следуют n строк, каждая из которых содержит непустую строку из строчных латинских букв. Гарантируется, что все строки различны.

Суммарная длина всех данных строк не превышает 10^5.

Выходные данные

Выведите непустую минимальную по длине хорошую строку, а из таких минимальную лексикографически, либо «NO» (без кавычек), если таких строк не существует.

Примеры

входные данные	Скопировать
4 mail ai lru cf	

выходные данные
cfmailru

входные данные	Скопировать
3 kek preceq cheburek	

выходные данные
NO

Примечание

Можно показать, что в первом тесте есть два варианта хорошей строки минимальной длины: «cfmailru» и «mailrucf». Первый вариант является минимальным лексикографически.

E. Максимум в массиве

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Как-то раз Петя решал очень интересную задачу. Однако, какие бы оптимизации Петя ни применял, его решение всё равно получало вердикт Time limit exceeded. После тщательного анализа кода он обнаружил, что проблема заключается в его функции поиска максимума в массиве из n положительных чисел, которая работала слишком долго. В отчаянии Петя решил добавить весьма неожиданное отсечение с параметром k, после чего функция приняла следующий вид:

```
int fast_max(int n, int a[]) {
    int ans = 0;
    int offset = 0;
    for (int i = 0; i < n; ++i)
        if (ans < a[i]) {
            ans = a[i];
            offset = 0;
        } else {
            offset = offset + 1;
            if (offset == k)
                return ans;
        }
    return ans;
}
```

Таким образом, функция последовательно перебирает элементы массива, поддерживая текущий максимум, при этом если после k последовательных итераций этот максимум не изменился, то текущий максимум возвращается как ответ.

Теперь Пете стало интересно, как часто его функция даёт неверный ответ. Он просит вас посчитать количество перестановок чисел от 1 до n таких, что результат работы его функции на этих перестановках не равен n. Так как ответ может быть большим, выведите его по модулю $10^9 + 7$.

Входные данные

В единственной строке находятся два целых числа n и k ($1 \le n, k \le 10^6$), разделённые пробелом — размер перестановок и параметр k соответственно.

Выходные данные

Выведите количество перестановок, на которых функция Пети выдаёт неправильный ответ, взятое по модулю $10^9 + 7$.

Примеры

входные данные	Скопировать
5 2	
выходные данные	
22	

входные данные	Скопировать
5 3	
выходные данные	
6	

входные данные	Скопировать
6 3	
выходные данные	
84	

Примечание

Искомые перестановки из второго примера:

$[4, 1, 2, 3, 5], [4, 1, 3, 2, 5], [4, 2, 1, 3, 5], [4, 2, 3, 1, 5], [4, 3, 1, 2, 5], [4, 3, 2, 1, 5]$.

F. Симметричные проекции

ограничение по времени на тест: 2 секунды
ограничение по памяти на тест: 256 мегабайт
ввод: стандартный ввод
вывод: стандартный вывод

Дано множество из n точек на плоскости. Назовем прямую, проходящую через начало координат, хорошей, если проекция данного множества точек на эту прямую образует симметричное мультимножество. Найдите количество хороших прямых.

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

Мультимножество точек называется симметричным, если существует такая точка P на плоскости, что данное мультимножество обладает центральной симметрией относительно точки P.

Входные данные

В первой строке дано целое число n ($1 \le n \le 2000$) — количество точек в данном множестве.

Каждая из следующих n строк содержит два целых числа x_i и y_i ($-10^6 \le x_i, y_i \le 10^6$), описывающие координаты точек множества. Гарантируется, что все точки различны.

Выходные данные

Если хороших прямых бесконечно много, выведите `-1`.

Иначе выведите одно целое число — количество хороших прямых.

Примеры

входные данные	Скопировать
3 1 2 2 1 3 3	

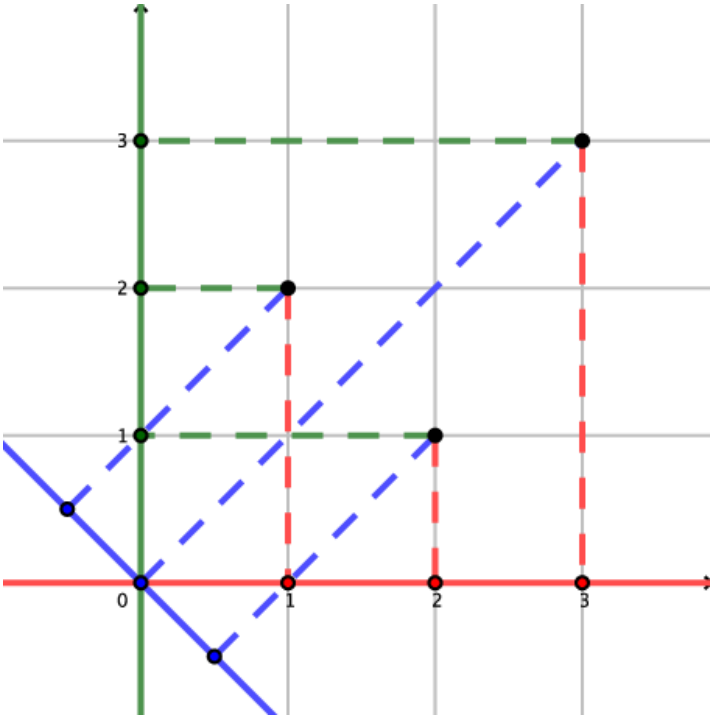
выходные данные
3

входные данные	Скопировать
2 4 3 1 2	

выходные данные
-1

Примечание

Иллюстрация к первому тесту из условия:



Во втором примере хорошей является любая прямая, проходящая через начало координат.
