

## Задача А. Перекрестные уги

Имя входного файла: `cross.in`  
Имя выходного файла: `cross.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта  
Отображение результатов:

Задан направленный ациклический раф  $G$ , содержащий  $n$  ершин и  $m$  уг. Параллельные уги в рафе отсутствуют. *Остовное поддерево* рафа  $G$  представляет собой ориентированное орневое ерево (уги направлены от орня), покрывающее все ершины  $G$ . Требуется выяснить, существует ли остовное поддерево рафа  $G$ , относительно которого в  $G$  нет перекрестных уг. (Напомним, что уга  $(x, y)$  рафа  $G$  называется *перекрестной* относительно остовного поддерева  $T$ , если  $x$  не достижима из  $y$  по  $T$  и наоборот.)

### Формат входных данных

В первой строке входного файла заданы числа  $n$  ( $2 \leq n \leq 100\,000$ ) и  $m$  ( $1 \leq m \leq 300\,000$ ). В последующих  $m$  строках описаны уги рафа. Каждая уга задается парой чисел — номерами начальной и конечной ершины соответственно. Ершины рафа нумеруются натуральными числами, начиная с 1.

### Формат выходных данных

В случае если искомого ерева не существует, то выведите в первой (и единственной) строке выходного файла число -1. Если искомое ерево  $T$  существует, то выведите в первой строке количество уг  $k$  в  $T$ . Во второй строке выведите  $k$  чисел — номера уг, образующих  $T$  (уги пронумерованы натуральными числами, начиная с 1 в соответствии с тем, как они заданы во входном файле). Номера уг необходимо выводить в порядке возрастания.

### Примеры

<code>cross.in</code>	<code>cross.out</code>
3 3 1 2 1 3 2 3	2 1 3
4 5 1 2 1 3 1 4 2 4 3 4	-1

## Задача В. Письма

Имя входного файла: `letter.in`  
Имя выходного файла: `letter.out`  
Ограничение по времени: 5 секунд  
Ограничение по памяти: 64 мегабайта  
Отображение результатов:

Расставаясь на летние каникулы, школьники Оксана и Петя договорились писать друг другу письма. Петя не отличается большой фантазией, поэтому каждое его письмо получается следующим образом. Вначале он задумывает  $n$  строк  $\alpha_1, \dots, \alpha_n$  ( $1 \leq n \leq 14$ ). Каждое письмо Пети представляет собой результат конкатенации (склеивания) этих строк, взятых в некотором порядке.

Одновременно, Петя отличается симпатичностью, поэтому его первое письмо представляет собой лексикографически минимальную возможную строку, получаемую описанным выше образом. Второе его письмо будет второй в лексикографическом порядке такой строкой. В общем случае  $k$ -е письмо Пети — это  $k$ -я в лексикографическом порядке строка. (В случае, если одна и та же строка может получаться конкатенацией  $\alpha_1, \dots, \alpha_n$  несколькими способами, то она учитывается соответствующее число раз.)

Оксана не уверена, что у Пети хватит терпения писать письма все каникулы, поэтому она хочет заранее знать, каковым будет  $k$ -е по счету письмо (письма нумеруются с 1). Помогите ей решить эту задачу!

### Формат входных данных

В первой строке входного потока заданы числа  $n$  и  $k$  ( $1 \leq k \leq n!$ ). В последующих  $n$  строках входного потока заданы строки  $\alpha_1, \dots, \alpha_n$ . Каждая строка представляет собой непустую последовательность из строчных латинских букв (без пробелов) длины не более 20.

### Формат выходных данных

Выведите в первой и единственной строке искомое  $k$ -е письмо Пети.

### Примеры

<code>letter.in</code>	<code>letter.out</code>
3 2 aa bb cc	aaccbb
3 2 aa a b	aaab

## Задача С. Списки

Имя входного файла: `list.in`  
Имя выходного файла: `list.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта  
Отображение результатов:

Вася получил на день рождения в подарок от Васи новенький двусвязный список. Список содержит  $N$  вершин, пронумерованных числами от 1 до  $N$ . Исходно эти числа расставлены по порядку слева направо.

Над списком разрешается производить два типа действий:

- А) Переместить вершину  $X$ , поместив ее непосредственно перед вершиной  $Y$ .
- В) Переместить вершину  $X$ , поместив ее непосредственно после вершины  $Y$ .

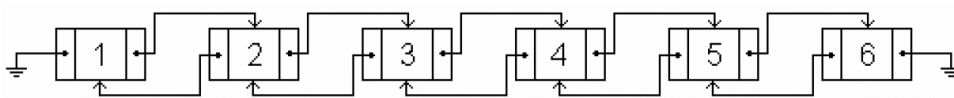


Рис. 1: Пример списка из 6 вершин

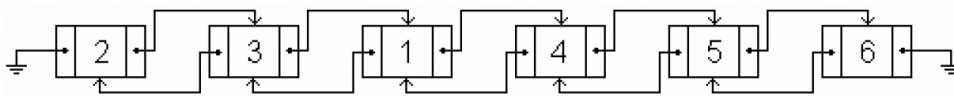


Рис. 2: Состояние списка после хода “А 1 4”

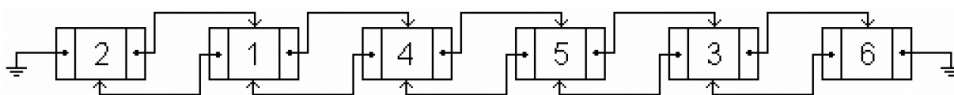


Рис. 3: Состояние списка после еще одного хода “В 3 5”

Вася играл со списком в течение нескольких часов, записывая каждый совершенный ход. Теперь его волнует такой вопрос: какова кратчайшая последовательность ходов, которая бы вернула список в его исходное состояние? Помогите Васе решить эту нелегкую задачу: напишите программу, которая по записям совершенных Васей ходов находит искомую кратчайшую последовательность.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $N$  и  $M$  ( $2 \leq N \leq 500\,000$ ,  $0 \leq M \leq 100\,000$ ) — количество элементов в списке и число выполненных Васей ходов соответственно. Каждая из последующих  $M$  строк описывает ходы Васи. Описание начинается с символа, обозначающего тип хода (“А” или “В”), за которым следуют числа  $X$  и  $Y$ .

### Формат выходных данных

На первой строке файла выведите искомое минимальное количество ходов  $K$ . В последующих  $K$  строках выведите описание ходов в том же формате, что и во входном файле.

В случае если число  $K$  и сама последовательность ходов правильные, ваша программа получит полный балл на данном тесте. Если же ваша программа выдаст правильное число  $K$ , но не выдаст последовательность ходов (либо выдаст неверную последовательность), то вы получите 60% от полного балла за данный тест.

## Примеры

list.in	list.out
2 1 A 2 1	1 A 1 2
4 3 B 1 2 A 4 3 B 1 4	2 A 1 2 B 4 3
6 5 A 1 4 B 2 5 B 4 2 B 6 3 A 3 5	3 A 4 5 B 6 5 A 2 3

## Задача D. Тройки

Имя входного файла: `triples.in`  
Имя выходного файла: `triples.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта  
Отображение результатов:

Задано простое число  $p$  и последовательность из  $n$  целых неотрицательных чисел:  $a_1, a_2, \dots, a_n$ .

Требуется найти количество троек чисел  $\langle i, j, k \rangle$ , таких что  $i \neq j$ ,  $i \neq k$ ,  $j \neq k$  и

$$a_i^{a_j} = a_k \pmod{p}.$$

Считается, что  $0^0 = 1$ .

### Формат входных данных

Первая строка входного файла содержит  $n$  и  $p$  ( $3 \leq n \leq 100\,000$ ,  $p \leq 1500$ ). Следующая строка содержит  $n$  целых неотрицательных чисел  $a_1, a_2, \dots, a_n$  ( $0 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите в выходной файл количество искомых троек.

### Пример

<code>triples.in</code>	<code>triples.out</code>
3 2 1 2 3	2