Задача 1. Перестановки

Имя входного файла: perm.in
Имя выходного файла: perm.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 Мбайт

Задана строка из n символов. Найдите все различные перестановки этой строки в лексикографическом порядке. Вам запрещается использование библиотеки <algorithm> на языке c++.

Формат входного файла

Входной файл содержит строку из n ($1 \le n \le 8$) символов.

Формат выходного файла

Выведите все возможные перестановки заданной строки в лексикографическом порядке.

Пример входных и выходных данных

perm.in	perm.out
AB	AB BA
212	122 212 221

Задача 2. Обратная перестановка

Имя входного файла:	reverse.in
Имя выходного файла:	reverse.out
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 Мбайт

Перестановкой из n элементов называется упорядоченный набор из n различных чисел от 1 до n. Обозначим перестановку в виде $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$, где σ_i — различные целые числа от 1 до n, то есть $\sigma_i \neq \sigma_j$ для любых $i \neq j$ и $1 \le \sigma_i \le n$. Число σ_i на i-ом месте перестановки запишем как значение функции: $\sigma(i) = \sigma_i$. Число n называется *порядком* перестановки.

Произведение перестановок σ и π вычисляется по следующему правилу:

$$\sigma * \pi = (\pi(\sigma(1)), \pi(\sigma(2)), \dots, \pi(\sigma(n))) = (\pi(\sigma_1), \pi(\sigma_2), \dots, \pi(\sigma_n)).$$

Тождественной перестановкой называется перестановка $\varepsilon = (1, 2, \dots, n)$.

Обратной к перестановке π называется такая перестановка π^{-1}, для которой выполняется $\pi * \pi^{-1} = \pi^{-1} * \pi = \varepsilon$, где ε – тождественная перестановка.

Вам необходимо по заданной перестановке найти обратную.

Формат входного файла

В первой строке записано одно число n ($0 < n \le 20\,000$) — порядок перестановки. Во второй строке записана сама перестановка.

Формат выходного файла

В выходной файл выведите обратную перестановку.

Пример входных и выходных данных

reverse.in	reverse.out
3 2 3 1	3 1 2

Задача 3. Тур по городу

Имя входного файла:	citytour.in
Имя выходного файла:	citytour.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 Мбайт

В Казани очень много красивых мест, которые стоит увидеть. Туристические агентства предлагают различные туры по этим местам. Одно из таких агентств предлагает за один тур посетить ровно m из этих достопримечательностей. Тур начинается и заканчивается в одном и том же месте. По пути из одного места до другого экскурсионный автобус может проехать мимо очередной достопримечательности, не посещая его, чтобы сократить маршрут.

Всего в городе n красивых мест, некоторые из которых связаны дорогами известной длины. От любого места можно добраться до другого.

Требуется написать программу, которая подберет наиболее короткий по длине тур, удовлетворяющий этим условиям.

Формат входного файла

Первая строка содержит три числа: n ($1 \le n \le 15$) — количество красивых мест в городе, m ($1 \le m \le 7$) – количество мест, которое необходимое посетить в туре, и k — количество дорог. Следующие k строк содержат по три целых числа. В i-ой из этих строк первые два числа соответствуют номерам мест, которые связывает i-ая дорога, третье число — ее длина d_i ($1 \le d_i \le 100$).

Формат выходного файла

Выведите длину кратчайшего тура.

Пример входных и выходных данных

citytour.in	citytour.out
5 4 7 1 5 3 5 4 2 4 3 3 3 1 10 1 4 15 2 3 5 2 1 3	16

Задача 4. Взлом шифра

Имя входного файла: cipher.in
Имя выходного файла: cipher.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 Мбайт

Кодовый замок содержит n кнопок, пронумерованных целыми числами от 1 до n. Замок открывается только тогда, когда какие-то последовательные n нажатий на кнопки образуют некоторую секретную перестановку чисел от 1 до n. Кнопки можно нажимать по очереди, нажимать одновременно две или более кнопок нельзя. Более формально: предположим, что вы нажали на кнопки k раз. Пусть a_i $(1 \le i \le k)$ — номер кнопки, которую вы нажали i-ой по счету, а $b_1, b_2, \ldots, b_n$ — секретная перестановка. Тогда замок открывается, если существует такое число m $(1 \le m \le k - n + 1)$, что $b_1 = a_m$, $b_2 = a_{m+1}, \ldots, b_n = a_{m+n-1}$.

Вам необходимо придумать такую универсальную последовательность нажатий, что при нажатии кнопок этой последовательности замок откроется для любой секретной перестановки. Эта последовательность должна содержать не более $2 \cdot n!$ чисел, где $n! = 1 \cdot 2 \cdot \ldots \cdot n$. Например, для $n = 3$ длина последовательности не должна превышать 12.

Требуется составить программу, которая для заданного n находит такую универсальную последовательность.

Формат входного файла

В единственной строке входного файла находится целое число n $(1 \le n \le 10)$ — количество кнопок на кодовом замке.

Формат выходного файла

В первой строке выходного файла выведите число k $(1 \le k \le 2 \cdot n!)$ — длину универсальной последовательности. Во второй строке выведите через пробел k целых чисел a_i $(1 \le a_i \le n)$, которые задают порядок нажатия кнопок. Достаточно вывести любую последовательность длины не более $2 \cdot n!$, минимизировать длину не нужно. Гарантируется, что такая последовательность существует для любого n.

Пример входных и выходных данных

cipher.in	cipher.out
2	3 1 2 1
3	10 1 2 3 1 3 2 1 3 1 2