

## Второе задание по практикуму для самостоятельного программирования дома

Напишите программу, которая получает на вход информацию об ориентированном графе и двух его вершинах и выводит длину кратчайшего пути, соединяющего эти вершины, и сам этот путь. В случае, когда кратчайших путей несколько, требуется вывести их все. В рассматриваемом графе из каждой вершины выходит не более одного ребра, кроме, возможно, особой вершины, число исходящих рёбер из которой не ограничено. Предполагается, что программа считывает информацию с клавиатуры.

Формат входной информации:

```
<Число вершин> <Номер одной вершины> <Номер другой вершины>
<Список вершин, в которые приводят рёбра, выходящие из первой вершины>
<Список вершин, в которые приводят рёбра, выходящие из второй вершины>
...
<Список вершин, в которые приводят рёбра, выходящие из последней вершины>
```

Гарантируется, что в списке вершин особой вершины все вершины различны. Если из вершины не выходят рёбра, то вместо соответствующего списка вершин вводится 0. Примеры ввода:

			5 1 3
2 1 2	3 1 3	3 1 2	2
2	2	0	3
1	3	2	4 5
	0	3 2 1	1
			1

Вывод программы должен производиться на экран и иметь следующий формат:

```
<Длина кратчайшего пути>
<Список вершин в первом кратчайшем пути>
<Список вершин во втором кратчайшем пути>
...
<Список вершин в последнем кратчайшем пути>
```

В списках вершины должны идти в порядке прихода в них, пути должны идти в лексикографическом порядке. Если пути, соединяющего вершины, нет, то должно быть выведено -1. Для примеров выше выводы *должны* быть следующими:

1			2
1 2	2	-1	1 2 3
2 1	1 2 3		3 4 1
			3 5 1

Требования:

1. В программе не должно быть явных априорных ограничений на количество вершин в графе и число вершин в списке особой вершины.
2. Вся выделенная программе динамическая память в конце её работы должна быть освобождена самой программой.
3. Запрещается использование функции `realloc` и массивов переменной длины (вида `int a[n];`, где `n` — переменная величина).
4. Количество выделенной динамической памяти должно быть  $O(N)$  (т.е. должно оцениваться сверху произведением константы, не зависящей от графа, на  $N$ ), где  $N$  — число вершин в графе. Затраты по времени работы программы также *должны* оцениваться как  $O(N)$ .

**Мягкий deadline: 10.11.2021.**

**Жёсткий deadline: 23:59, 17.11.2021.**