**ЛАБОРАТОРНАЯ РАБОТА № 14**

**АЛГОРИТМЫ НА ГРАФАХ**

## ТЕОРИЯ И ПОЯСНЕНИЯ

В работе рекомендуется проанализировать алгоритмы, приведенные в учебном пособии Ю.Л. Костюка «Лекции по основам программирования», и применить их при разработке программ.

## ЗАДАНИЯ

### ПРЕДСТАВЛЕНИЕ ГРАФОВ В ПАМЯТИ

**Упражнение 1.** Неориентированный граф задан произвольной последовательностью ребер. Исходные данные находятся в текстовом файле. Создайте матрицу смежности для неориентированного графа. Выведите результат в отдельный текстовый файл.

**Упражнение 2.** Используя данные из текстового файла, полученного при выполнении упр. 1, постройте массив номеров смежных вершин для неориентированного графа. Выведите результат в отдельный текстовый файл.

**Упражнение 3.** Используя данные из текстового файла, полученного при выполнении упр. 2, сформируйте последовательность ребер неориентированного графа. Выведите результат в отдельный текстовый файл.

Все упражнения следует выполнить в рамках одного проекта.

### ГЕНЕАЛОГИЯ

В рамках социологического исследования был создан пронумерованный список жителей города и собраны сведения о родственных отношениях жителей города. Эти сведения представлены в виде набора пар номеров людей, являющихся ближайшими родственниками: муж – жена, родитель – ребенок. Напишите программу, которая определяет, сколько непересекающихся генеалогических деревьев можно построить на заданном наборе пар родственников.

***Входные величины:*** набор пар номеров жителей города загружается из текстового файла. Последняя пара: -1 -1.

***Результат:*** в текстовый файл выводится количество непересекающихся генеалогических деревьев и перечень номеров жителей города, относящихся к каждому генеалогическому древу.

### ПУТЕШЕСТВИЯ

На рисунке – схема дорог, связывающих города А, Б, В, Г, Д, Е, Ж, И, К, Л. По каждой дороге можно двигаться только в одном направлении, указанном стрелкой. Сколько существует различных маршрутов из города А в город Л?

**Б**

**Г**

**И**

**В**

**Е**

**Л**

**А**

**Ж**

**К**

**Д**

***Входные величины:*** данные, представляющие схему дорог, загружаются из текстового файла.

***Результат:*** количество маршрутов выводится на экран.

### ВОСЬМИПРОХОДНЫЙ ЛАБИРИНТ

Задан лабиринт в виде прямоугольной таблицы. Клетка таблицы со значением ‘.’ (или ‘-‘, но не пробел) соответствует проходу в лабиринте, клетка со значением ‘#’ – стена. Проход из клетки в клетку возможен по восьми направлениям: направо, налево, вверх, вниз, по диагонали вверх направо, по диагонали вверх налево, по диагонали вниз направо, по диагонали вниз налево. Начало пути по лабиринту отмечается символом ‘s’, а конец – символом ‘f’. Используя метод просмотра графа в ширину, найдите кратчайший путь в лабиринте или определите, что пути не существует.

***Входные величины:*** таблица, представляющая лабиринт, загружается из текстового файла. Для упрощения проверок рекомендуется задавать стены по границам лабиринта.

***Результат:*** та же таблица, в ней символом ‘\*’ отмечены клетки, по которым проходит кратчайший путь. Таблица выводится на экран. Если пути не существует, то выдается соответствующее сообщение.

### ДЕТСТВО И ГРАФЫ

Напишите программу, которая проверяет возможность нарисовать фигуры, подобные изображенным на рисунках, не отрывая карандаша от бумаги, причем каждую линию фигуры карандаш должен проходить только один раз.

***Входные величины:*** данные, представляющие рисунок (одна фигура), загружаются из текстового файла.

***Результат:*** вывести результат проверки.

* + - 1. Можно нарисовать эту фигуру, начиная и заканчивая рисование в одной и той же точке.
      2. Можно нарисовать эту фигуру, начиная и заканчивая рисование в различных точках.
      3. Нельзя нарисовать эту фигуру (без наложения линий), не отрывая карандаша от бумаги.

Рис. 1 Рис. 2