**Теория. Ввод Вывод в С++. Файловый ввод/вывод 14.5.2021 г.**

**Понятие потока. Стандартные потоки ввода/вывода. Форматирование данных при вводе/выводе – флаги, функции, манипуляторы*.***

**Использование файлов в программах. Ввод/вывод текстовых файлов. Использование операций << и >>. Посимвольный ввод/вывод. Построчный ввод/вывод. Произвольный доступ к файлу.**

Сделать надо **одну** задачу**!!!** Сдача: до **25.5.2021 года (**включительно**)**

**Задачи**.

1. Исходную строку **str** записать в файл "**data1**". Скопировать файл "**data1**" (не считывая его полностью в **ОП** компьютера) в файл "**data2**", при этом удалить лишние пробелы.
2. В каждой строке матрицы **A(n x m)** найти и удалить максимальные элементы.

В программе использовать **динамическую матрицу**. Исходная матрица должна быть прочитана из текстового файла, структура файла следующая – число строк, число столбцов, а затем идут элементы матрицы. Результат работы программы вывести на консоль и в текстовый файл.

Все действия, повторяющиеся в программе два и более раза, должны быть оформлены в функцию.

1. Элементы массива А(N) упорядочить в порядке возрастания, используя сортировку **Шелла** (**см**. ниже), и вставить ***k*** заданных чисел, не нарушая упорядоченности массива.

Исходный массив должен быть прочитан из текстового файла. Результат работы программы вывести на консоль и в текстовый файл.

**Сортировка Шелла**

Основная идея сортировки Шелла состоит в том, что на начальном этапе реализуется сравнивание и, если требуется, перемещение далеко отстоящих друг от друга элементов. Интервал между сравниваемыми элементами (**step**) постепенно уменьшается до единицы, что приводит к перестановке соседних элементов на последних стадиях сортировки.

Реализуем метод Шелла следующим образом. Начальный шаг сортировки примем равным **n/2**, т.е. **1/2** от общей длины массива, и после каждого прохода будем уменьшать его в два раза. Каждый этап сортировки включает в себя проход всего массива и сравнение отстоящих на **step** элементов. Проход с тем же шагом повторяется, если элементы переставлялись. Заметим, что после каждого этапа отстоящие на **step** элементы отсортированы.

Пусть требуется рассортировать массив: 41,53,11,37,79,19, 7,61.

В строке после массива в круглых скобках указаны индексы сравниваемых элементов и указан номер внешнего цикла.

41 53 11 37 79 19 7 61 – исходный массив

41 19 11 37 79 53 7 61 – (0,4), (1,5) 1-ый цикл

41 19 7 37 79 53 11 61 – (2,6), (3,7) 1-ый цикл

7 19 41 37 11 53 79 61 – (0,2), (1,3), (2,4), (3,5) (4,6), (5,7) 2-ой цикл

7 19 11 37 41 53 79 61 – (0,2), (1,3), (2,4), (3,5) (4,6), (5,7) 2-ой цикл

7 11 19 37 41 53 61 79 – сравнивались соседние элементы 3-ий цикл