

---

Примеры программ для экзамена  
по курсу  
Основы ЭВМ и программирование  
механико-математический факультет  
2 курс, 3 поток, 2020 учебный год

Написать с использованием STL реализацию хеш-таблицы на основе метода линейных проб множества строк длины не более 256 (от STL здесь используется реализация вектора и итераторов; при этом придется написать свою реализацию простого (!) класса строк длины не более 256). Хеш-функцию реализовать произвольным образом. Заполнить таблицу словами из файла

<http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку; длину таблицы задать как двойное количество слов во входном файле). Для каждого значения  $i$  хеш-функции вывести все слова из соответствующего списка в файл  $i.txt$  по одному слову в строке (в имени файла  $i$  – текстовое представление соответствующего числа).

Написать программу на языке C, которая в изображении, заданном в файле <http://lectures.stargeo.ru/1.bmp> (изображение с палитрой) заменяет все пиксели с красным цветом, имеющие синих соседей, на пиксели с зеленым цветом. Предполагается, что указанные цвета в палитре есть. Результат вывести в файл `./1.bmp`. Заголовок BMP-файла можно взять из файла <http://lectures.stargeo.ru/header/bmp.h>.

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла <http://lectures.stargeo.ru/t0.txt> в массив строк (строки длиной не более 256 символов), лексикографически сортирует массив строк с помощью встроенной функции `qsort` и выводит полученный массив в файл `./t0.txt` по одной строке в строку файла.

На языке C++ написать собственную ссылочную реализацию двунаправленного списка строк (строки длиной не более 256 символов). Последовательно заполнить список словами из файла

<http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку). После считывания всех строк вывести в обратном порядке все элементы списка в файл `./t0.txt` по одному слову в строке (таким образом, все слова из исходного файла должны быть выведены в выходной файл, но в обратном порядке).

Написать на языке Python функцию, реализующую метод Гаусса решения системы линейных уравнений. Матрицу задавать как словарь, индексированный кортежами. Протестировать написанную функцию.

Написать программу на языке C, которая погружает изображения, заданное в файле `./1.bmp` (изображение с палитрой), в черную рамку шириной 2

пикселя. При этом ширина и высота изображения должны увеличиться, соответственно, на 4 пикселя. Результат вывести в файл `./1.bmp`. Заголовок BMP-файла можно взять из файла

<http://lectures.stargeo.ru/header/bmp.h> .

Написать функцию `double Deg2N(double x,int n)`; умножающую число  $x$  типа `double` на множитель  $2^n$ , использующую только сложение степени в представлении числа с числом  $n$ .

В файле

<http://lectures.stargeo.ru/v.txt> заданы координаты вершин графа по паре координат в каждой строке. В каждой строке файла

<http://lectures.stargeo.ru/e.txt> заданы номера вершин очередного ребра графа и длина данного ребра. С помощью STL реализовать алгоритм Дейкстры (модификацию алгоритма Дейкстры для STL), находящий кратчайший путь в графе (вершины и ребра графа можно загружать из указанных файлов) от первой до последней вершины из списка. Список вершин найденного пути и длину пути вывести в файл `./rez.txt` .

Реализовать на языке C хеш-функцию на основе умножения. Создать массив целых чисел, в котором в элементе с индексом  $i$  будет храниться количество слов из файла <http://lectures.stargeo.ru/t0.txt> (слова длиной не более 256 символов, разделяющиеся пробелами, табуляциями и переходами на следующую строчку), которым соответствует значение хеш-функции  $= i$ . Вывести на экран значения данного массива, среднее квадратичное отклонение значений данного массива и нормированное среднее квадратичное отклонение значений данного массива ( $D/M$ ).

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла

<http://lectures.stargeo.ru/t0.txt> в массив строк (строки длиной не более 256 символов), лексикографически сортирует массив строк методом быстрой сортировки делением пополам без рекурсии и выводит полученный массив в файл `./t0.txt` по одной строке в строку файла.

На языке C++ написать собственную реализацию очереди строк (строки длиной не более 256 символов, количество элементов в очереди не более 20). Последовательно заполнять очередь словами из файла

<http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку) и при заполнении очереди выводить все элементы очереди в файл `./t0.txt` по одному слову в строке (таким образом, все слова из исходного файла должны быть выведены в выходной файл).

Написать программу на языке C, которая транспонирует матрицу изображения, заданного в файле

<http://lectures.stargeo.ru/1.bmp> (изображение с палитрой). Результат вывести в файл `./1.bmp`. Заголовок BMP-файла можно взять из файла

<http://lectures.stargeo.ru/header/bmp.h> .

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла <http://lectures.stargeo.ru/t1.txt> в массив строк (строки длиной не более 10 символов и состоят только из цифр), лексикографически сортирует массив строк с помощью цифровой сортировки с использованием сортировки подсчетом и выводит полученный массив в файл `./t1.txt` по одной строке в строку файла.

На основе STL создать вектор из K списков целых чисел. Заполнить списки числами, записанными в файле <http://lectures.stargeo.ru/t2.txt> таким образом, чтобы число, остаток от деления которого на K был бы равен i, заносился бы в список с индексом i. Число K должно вводиться с клавиатуры. Содержимое каждого списка должно быть выведено в отдельную строку файла `./t2.txt` (через пробел). При выводе перебирать элементы вектора и списков надо через соответствующие итераторы.

Написать на языке C++ функцию, реализующую метод Гаусса решения системы линейных уравнений. Матрицу задавать как указатель на указатель. Размерность матрицы передавать через параметры вместе с самой матрицей и правой частью уравнения. Ввести матрицу в указанном виде из файла, где она записана естественным образом (по одной строке матрицы в строке файла без отдельного указания размера матрицы). Пусть правая часть СЛУ задается в том же файле дополнительным столбцом данной матрицы. Решить СЛУ созданной функцией. Результат вывести на экран.

Написать программу на языке C, которая погружает изображения, заданное в файле <http://lectures.stargeo.ru/32.bmp> (изображение в формате True Color), в черную рамку шириной 2 пикселя. При этом ширина и высота изображения должны увеличиться, соответственно, на 4 пикселя. Результат вывести в файл `./32.bmp`. Заголовок BMP-файла можно взять из файла <http://lectures.stargeo.ru/header/bmp.h>.

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла <http://lectures.stargeo.ru/t0.txt> в массив строк (строки длиной не более 256 символов), и находит k-ую порядковую статистику для данного массива (в смысле лексикографического сравнения) алгоритмом, обеспечивающим поиск порядковой статистики за линейное время в среднем. Полученное слово вывести на экран.

Написать с использованием STL реализацию хеш-таблицы на основе метода многих списков множества строк длины не более 256 (при этом придется написать свою реализацию простого (!) класса строк длины не более 256). Хеш-функцию реализовать произвольным образом. Заполнить таблицу словами из файла <http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку). Для каждого значения i хеш-функции вывести все слова из соответствующего списка в файл `i.txt` по одному слову в строке (в имени файла i – текстовое представление соответствующего числа).

Реализовать функцию расчета CRC-значения для строки, дополненной двумя нулевыми байтами. Занести в два последних нулевых байта найденное CRC-значение и проверить, что у полученной строки CRC-значение =0 .

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла <http://lectures.stargeo.ru/t0.txt> в массив строк (строки длиной не более 256 символов), лексикографически сортирует массив строк методом быстрой сортировки QSort и выводит полученный массив в файл `./t0.txt` по одной строке в строку файла. Можно использовать любую собственную реализацию алгоритма QSort.

На языке C++ написать собственную реализацию однонаправленного списка строк с фиктивным элементом (строки длиной не более 256 символов). Последовательно заполнить список словами из файла <http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку). После считывания всех строк вывести все элементы списка в файл `./t0.txt` по одному слову в строке (таким образом, все слова из исходного файла должны быть выведены в выходной файл).

Написать программу на языке Python, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла <http://lectures.stargeo.ru/t0.txt> в список строк, лексикографически сортирует список стандартной функцией и выводит полученный массив в файл `./t0.txt` по одной строке в строку файла.

Написать программу на языке C, которая транспонирует матрицу изображения, заданного в файле <http://lectures.stargeo.ru/32.bmp> (изображение в формате True Color). Результат вывести в файл `./32.bmp`. Заголовок BMP-файла можно взять из файла <http://lectures.stargeo.ru/header/bmp.h> .

Написать функцию `float Deg2N(float x,int n)`; умножающую число  $x$  типа `float` на множитель  $2^n$ , использующую только сложение степеней в представлении числа с числом  $n$ .

Написать собственную реализацию двунаправленного списка строк (строки длиной не более 256 символов) на основе двух стеков, реализованных с помощью STL. Последовательно заполнить список словами из файла <http://lectures.stargeo.ru/t0.txt> (слова разделяются пробелами, табуляциями и переходами на следующую строчку). После считывания всех строк вывести в обратном порядке все элементы списка в файл `./t0.txt` по одному слову в строке (таким образом, все слова из исходного файла должны быть выведены в выходной файл, но в обратном порядке).

Реализовать на языке C хеш-функцию на основе деления. Создать массив целых чисел, в котором в элементе с индексом  $i$  будет храниться количество слов из файла

*http://lectures.stargeo.ru/t0.txt* (слова длиной не более 256 символов, разделяющиеся пробелами, табуляциями и переходами на следующую строчку), которым соответствует значение хеш-функции =  $i$ . Вывести на экран значения данного массива, среднеквадратичное отклонение значений данного массива и нормированное среднеквадратичное отклонение значений данного массива (D/M).

Написать программу на языке C, которая вводит все слова, разделенные пробелами, табуляциями и переходами на следующую строчку, из файла *http://lectures.stargeo.ru/t0.txt* в массив строк (строки длиной не более 256 символов), лексикографически сортирует массив строк методом быстрой сортировки делением пополам с рекурсией и выводит полученный массив в файл *./t0.txt* по одной строке в строку файла.

На языке C++ написать собственную реализацию стека строк (строки длиной не более 256 символов) неограниченной длины на базе вектора (с реаллокацией памяти в случае заполнения стека). Заполнить стек словами из файла *http://lectures.stargeo.ru/t0.txt* (слова разделяются пробелами, табуляциями и переходами на следующую строчку). Вывести все элементы стека в файл *./t0.txt* по одному слову в строке.

Написать с использованием STL реализацию хеш-таблицы на основе метода линейных проб множества строк длины не более 256 (от STL здесь используется реализация вектора и итераторов; при этом придется написать свою реализацию простого (!) класса строк длины не более 256). Хеш-функцию реализовать произвольным образом. Заполнить таблицу словами из файла *http://lectures.stargeo.ru/t0.txt* (слова разделяются пробелами, табуляциями и переходами на следующую строчку; длину таблицы задать как двойное количество слов во входном файле). Для каждого значения  $i$  хеш-функции вывести все слова из соответствующего списка в файл *i.txt* по одному слову в строке (в имени файла  $i$  – текстовое представление соответствующего числа).