

Лабораторная работа № 5. Базовые элементы STL

std::string, std::set, std::multiset, std::pair, std::map, std::multimap, auto, range-based for loop, алгоритмы: sort, find, count, for_each

Указание.

1. Для работы с массивами рекомендуется использовать контейнеры *std*.
2. Для работы с файлами познакомьтесь с материалами:
<https://edufpmi.bsu.by/mod/resource/view.php?id=949> ,
<https://metanit.com/cpp/tutorial/8.1.php>
3. Ввод и вывод исходных данных оформляйте с помощью функций. Размещайте эти функции в отдельных файлах.
4. Для всех функций разработайте unit-тесты.

Познакомьтесь с документацией Google Test Framework Gtest <https://github.com/google/googletest/tree/master/googletest/docs> , а также изучите примеры <https://github.com/google/googletest/blob/master/googletest/docs/primer.md> и образцы проектов с использованием gtest <https://github.com/google/googletest/blob/master/googletest/docs/samples.md> При разработки тестов используйте разнообразные ASSERT_*/EXPECT_*.

Чтение и запись в файл – пример кода:

```
#include <iostream>
#include <fstream>
#include <string>

using std::ofstream;
using std::ifstream;
using std::endl;
using std::cout;
using std::string;
using std::ios;

int main() {
    // Папка data должна существовать в папке проекта
    string filePath = "data/dataOut.txt";
    ofstream output(filePath, ios::app);
    int n = 10;

    for (int i = 0; i < n; ++i) {
        output << i + 1 << ' ';
    }
    output.close();

    filePath = "data/data.txt";
    string str;
    ifstream fin(filePath, ios::in);
    while (getline(fin, str)) {
        cout << str << endl;
    }
    fin.close();
    system("pause");
    return 0;
}
```

Задание.

Разработайте приложения для решения следующих задач:

Задачи на семинар:

1. Даны два множества A и B (`std::multiset`). Найти объединение ($A \cup B$), пересечение ($A \cap B$), разность этих множеств ($A \setminus B$, $B \setminus A$).
2. Во входном файле строки из маленьких латинских букв, вывести количество различных строк.
3. Дан текстовый файл. Построить частотный словарь слов этого текста. Знаки препинания в словах игнорировать (`std::map`).
4. Дан вектор пар строк. Для каждого элемента вектора вывести YES или NO если вторая строка является подстрокой первой и позицию первого вхождения (или -1, если NO).

Задачи на семинар:

5. Дано 2 файла, один с исходными данными (числа, M штук), второй файл с N запросами (каждый запрос в отдельной строке): `add number` (например, `add 7`) - добавить число, `delete number` - удалить число, `find_min` - вывести минимум, `is_exist number` - проверить принадлежность коллекции. Загрузить исходные данные в коллекцию и выполнить над элементами все запросы из второго файла. Результат выполнения записать в выходной файл. (`std::multiset`)
6. Есть N задач, каждая выполняется раз в $a[i]$ секунд. На секунде 0 выполнились все задачи (то есть первая будет в 0, $a[1]$, $2*a[1]$...). Вывести K первых выполнений. Если две задачи в одну и ту же секунду выполняются, то первой вывести с меньшим номером. (`std::pair<int, int>`)
7. Дано N пар строк из маленьких латинских букв - название улицы и фамилия человека. Нужно для каждой улицы вывести сколько различных фамилий на ней живет. Улицы выводить в лексикографическом порядке (`map<string, set<string>>` или `vector<pair<string, string>>` и `std::unique`)
8. Даны N чисел. N - 1 раз делают следующее - берут два минимальных числа, выкидывают их из массива и добавляют их сумму. На каждом шаге алгоритма выводите минимум и максимум в массиве.