

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Кафедра Інформатики

Звіт  
з лабораторної роботи №6  
з дисципліни «Об'єктно-орієнтоване програмування»

Варіант 12

Виконав:  
ст. гр. ІТІНФ-21-1  
Кошель В.О.

Перевірили:  
Кириченко І.Ю.

## **6. АСОЦІАТИВНІ КОНТЕЙНЕРИ**

### **6.1. Мета роботи:**

Ознайомитись з такими поняттями, як асоціативні контейнери стандартної бібліотеки STL. Отримати навички роботи з ними.

### **6.2. Завдання:**

Розробити програму, яка зчитує текст і зберігає номери всіх рядків, в яких зустрілося дане слово, за винятком слів, зазначених у списку.

Слова в тексті розділяються пробілами.

Накопичення номерів рядків реалізовувати за допомогою асоціативного масиву, список слів для виключення реалізовувати за допомогою множини.

Кожен раз, коли зустрілося слово зі списку, виводити повідомлення.

Якщо слово зустрілося в одному рядку двічі - виводити повідомлення.

Створити набір функцій для виведення результатів і винести їх в окремий файл.

Оголошення функцій винести в заголовочний файл. Вихідні дані зчитати з файлу і вивести в файл

### **6.3. Вхідні дані (*input.txt*)**

уж небо осенью дышало дышало  
уж реже солнышко блистало  
короче становился день  
лесов таинственная сень

### **6.4. Слова-винятки (*excepts.txt*)**

день , осень , лесов , короче , Пушкин

## 6.5. Хід виконання:

### Файл Source.cpp

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
#include <map>
#include <set>
#include <algorithm>
#include "funcs.h"

using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    ofstream f_out("output.txt");

    map<string, multiset<size_t>> result; // Асоціативний контейнер результатів
    set<string> excepts;                // Множина слів-винятків

    /*
        Зчитування слів-винятків
    */
    ifstream f_excepts("excepts.txt");
    while (!f_excepts.eof())
    {
        string tmp;
        getline(f_excepts, tmp, ',');
        excepts.emplace(tmp);
    }
    f_excepts.close();

    /*
        Зчитування тексту
    */
    ifstream f_text("input.txt");
    string tmp;
    for (size_t i = 1; !f_text.eof(); i++)
    {
        getline(f_text, tmp, '\n');
        istringstream tmp_stream(tmp);
        while (!tmp_stream.eof())
        {
            string tmp_string;
            getline(tmp_stream, tmp_string, ' ');

            // Якщо знайдене слово не належить до множини винятків,
            // то додаємо до масиву пару <слово, рядок>
            if (excepts.find(tmp_string) == excepts.end())
                result[tmp_string].emplace(i);
            // Кожен раз, коли зустрілося слово зі списку, виводити повідомлення
            else
                Out::is_in_exception_list(f_out, tmp_string, i);
        }
    }

    map<string, set<size_t>> how_many_double;
    for (auto elem : result)
    {
        for (size_t i : elem.second)
        {
```

```

        size_t count_double = count(elem.second.begin(), elem.second.end(), i);

        if (count_double == 2)
            how_many_double[elem.first].emplace(i);
    }

    // Якщо слово зустрілося в одному рядку двічі - виводити повідомлення
    cout << "\nДва рази:\n";
    f_out << "\nДва рази:\n";
    for (auto elem : how_many_double)
        Out::is_double(f_out, elem.first, elem.second);

    return 0;
}

```

### Файл funcs.h

```

#pragma once
#include <iostream>
#include <string>
#include <fstream>
#include <set>

using namespace std;

namespace Out
{
    void is_in_exception_list(ofstream& file, string str, size_t line);
    void is_double(ofstream& file, string word, set<size_t> lines);
}

```

### Файл funcs.cpp

```

#include "funcs.h"

void Out::is_in_exception_list(ofstream& file, string str, size_t line)
{
    cout << "\"" << str << "\" знаходиться в списку виключень. В тексті - на " << line << " строці.\n";
    file << "\"" << str << "\" знаходиться в списку виключень. В тексті - на " << line << " строці.\n";
}

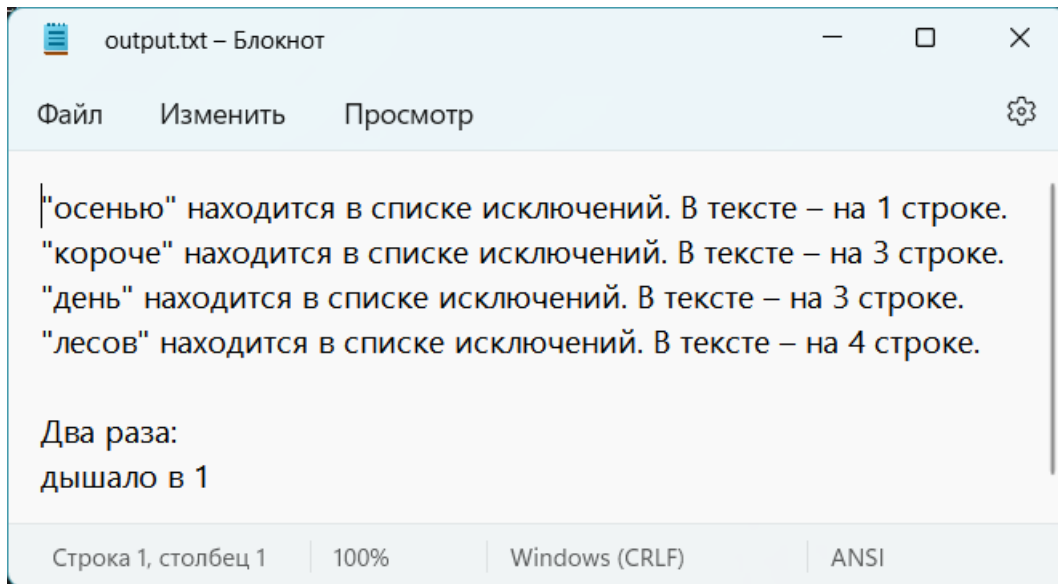
void Out::is_double(ofstream& file, string word, set<size_t> lines)
{
    cout << word << " y ";
    for (size_t i : lines)
        cout << i << "\t";
    cout << "\n";

    file << word << " y ";
    for (size_t i : lines)
        file << i << "\t";
    file << "\n";
}

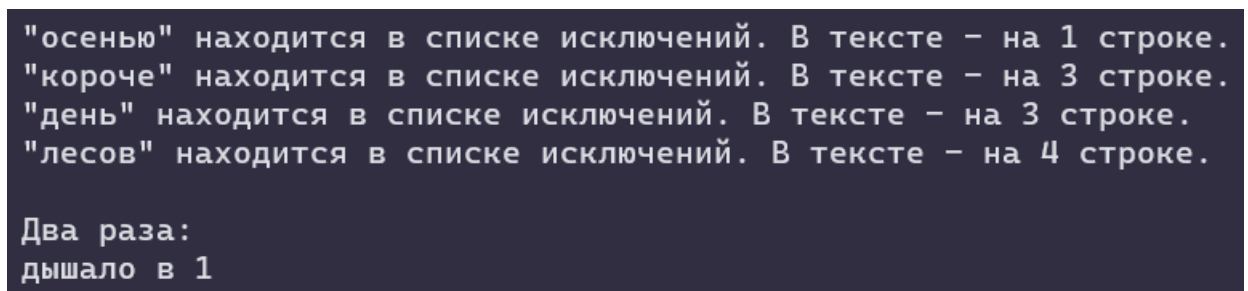
```

## 6.6. Результат виконання

- **Вихідний файл (*output.txt*):**



- **Консоль:**



6.7. **Висновок.** В результаті виконання цієї лабораторної роботи ми ознайомилися із асоціативними контейнерами стандартної бібліотеки та навчилися використовувати отримані знання на практиці.