Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 1 з дисципліни
«Основи програмування-2.
Методології програмування»

«Файли даних. Текстові файли»

Варіант 28

Виконав студент ІП-11 Сідак Кирил Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2022

Лабораторна робота №1

Мета: вивчити особливості створення і обробки текстових файлів даних.

Варіант 28

Створити текстовий файл. Сформувати новий текстовий файл, що складається з слів вхідного файлу, які зустрічаються у ньому менше N раз. Розмістити ці слова в новому файлі в порядку спадання їхньої довжини. Вивести вміст вихідного і створеного файлів.

Поставка задачі:

За умовою задачі треба створити текстовий файл, відкрити його та записати в нього текст, який ввів користувач з клавіатури. Далі потрібно створити список, у який додати усі слова, що містяться в цьому файлі, потім прибрати слова, що зустрічаються більше, ніж N раз, та відсортувати цей список за спаданням довжини слів. Далі треба записати елементи цього списку у новий файл та вивести вміст обох файлів.

Програма на C++

main.cpp

```
#include "file_operations.h"

int main() {
    int n;
    string first_file =
"/Users/kyryl/Desktop/Lab_1_C++/file_1.txt";
    string second_file =
"/Users/kyryl/Desktop/Lab_1_C++/file_2.txt";
    cout << "Enter N:" << endl;
    cin >> n;
    string file_mode = enter_file_mode();
    cout << "Terminate input with Command + D. Input for the
file:" << endl;
    create_first_file(first_file, file_mode);
    auto words_list = get_words_list(first_file, n);
    cout << "Sorted list by word length descending:" << endl;
    output_vector(words_list);
    create_second_file(second_file, words_list);
    cout << "First file:" << endl;
    output_file(first_file);
    cout << "Second file:" << endl;
    output_file(second_file);
    return 0;
}
```

file_operations.h

```
#ifndef LAB_1_C___FILE_OPERATIONS_H
#define LAB_1_C___FILE_OPERATIONS_H
#include <iostream>
```

```
#include <fstream>
#include <string>
#include <vector>
using namespace std;

string enter_file_mode();
void create_first_file(string, string);
vector<string> get_words_list(string, int);
void create_second_file(string, vector<string>);
void output_file(string);
void output_vector(vector<string>);
#endif
```

file_operations.cpp

```
#include "file_operations.h"
#include "string_functions.h"


string enter_file_mode() {
    string file_mode;
    cout << "Do you want to overwrite the file or append
input to it? Enter w or a:" << endl;
    cin >> file_mode;
    while (file_mode != "w" and file_mode != "a") {
        cout << "Incorrect input. Enter 'w' or 'a'." << endl;
        cout << "Do you want to overwrite the file or append
input to it? Enter w or a:" << endl;
        cin >> file_mode;
    }
    return file_mode;
}


void create_first_file(const string file_name, string mode) {
    ofstream file;
    if (mode == "w") {
        file.open(file_name);
    }
    else {
        file.open(file_name, ios::app);
    }
    string text;
    while (getline(cin, text))
        if (text.length() > 0) {
            file << text << endl;
        }
    file.close();
}


vector<string> get_words_list(string file_name, int n) {
    vector<string> words_list;
```

```
    string text;
    ifstream file(file_name);
    while (!file.eof()) {
        getline(file, text);
        vector<string> words = split(text);
        for (int j = 0; j < words.size(); ++j) {
            words_list.push_back(words[j]);
        }
    }
    file.close();
    vector<string> new_words_list;
    for (int i = 0; i < words_list.size(); ++i) {
        string word = words_list[i];
        int word_count = count(words_list.begin(),
words_list.end(), word);
            if (word_count < n) {
                if (!count(new_words_list.begin(),
new_words_list.end(), word)) {
                    new_words_list.push_back(word);
                }
            }
        }
    cout << "List of words which occur less than " << n << "
times:" << endl;
    output_vector(new_words_list);
    sort_desc(new_words_list);
    return new_words_list;
}


void create_second_file(string file_name, vector<string>
words) {
    ofstream file(file_name);
    auto size = words.size();
    for (int i = 0; i < size; ++i) {
        if (i != size - 1) file << words[i] << endl;
        else file << words[i];
    }
    file.close();
}


void output_file(string file_name) {
    ifstream file(file_name);
    string text;
    while (!file.eof()) {
        getline(file, text);
        cout << text << endl;
    }
    file.close();
}
```

```
void output_vector(vector<string> vect) {
    for (string str: vect)
        cout << str << ' ';
    cout << endl;
}
```

string_functions.h

```
#ifndef LAB_1_C___STRING_FUNCTIONS_H
#define LAB_1_C___STRING_FUNCTIONS_H
#include <string>
#include <vector>
using namespace std;
vector<string> split(string);
void sort_desc(vector<string>&);
#endif
```

string_functions.cpp

```
#include "string_functions.h"

vector<string> split(string str) {
    vector<string> words;
    string temp_str;
    for (int i = 0; i < str.length(); ++i) {
        if (str[i] == ' ' || i == str.length() - 1) {
            if (str[i] == ' ') {
                words.push_back(temp_str);
                temp_str = "";
            } else {
                temp_str += str[i];
                if (!temp_str.empty())
words.push_back(temp_str);
            }
        } else {
            temp_str += str[i];
        }
    }
    return words;
}

void sort_desc(vector<string>& vect) {
    string temp;
```

```
    for (int i = 0; i < vect.size() - 1; ++i) {
        for (int j = 0; j < vect.size() - 1 - i; ++j) {
            if (vect[j].length() < vect[j+1].length()) {
                temp = vect[j];
                vect[j] = vect[j+1];
                vect[j+1] = temp;
            }
        }
    }
}
```

Програма на Python

main.py

```
from file_operations import *


def main():
    first_file = 'file_1.txt'
    second_file = 'file_2.txt'
    n = int(input('Enter N: '))
    file_mode = enter_file_mode()
    print('Terminate input with Command + D. Input for the
file:')
    create_first_file(first_file, file_mode)
    words_list = get_words_list(first_file, n)
    print(f'Sorted list by word length descending:
{words_list}')
    create_second_file(second_file, words_list)
    print('First file:')
    output_file(first_file)
    print('Second file:')
    output_file(second_file)


if __name__ == '__main__':
    main()
```

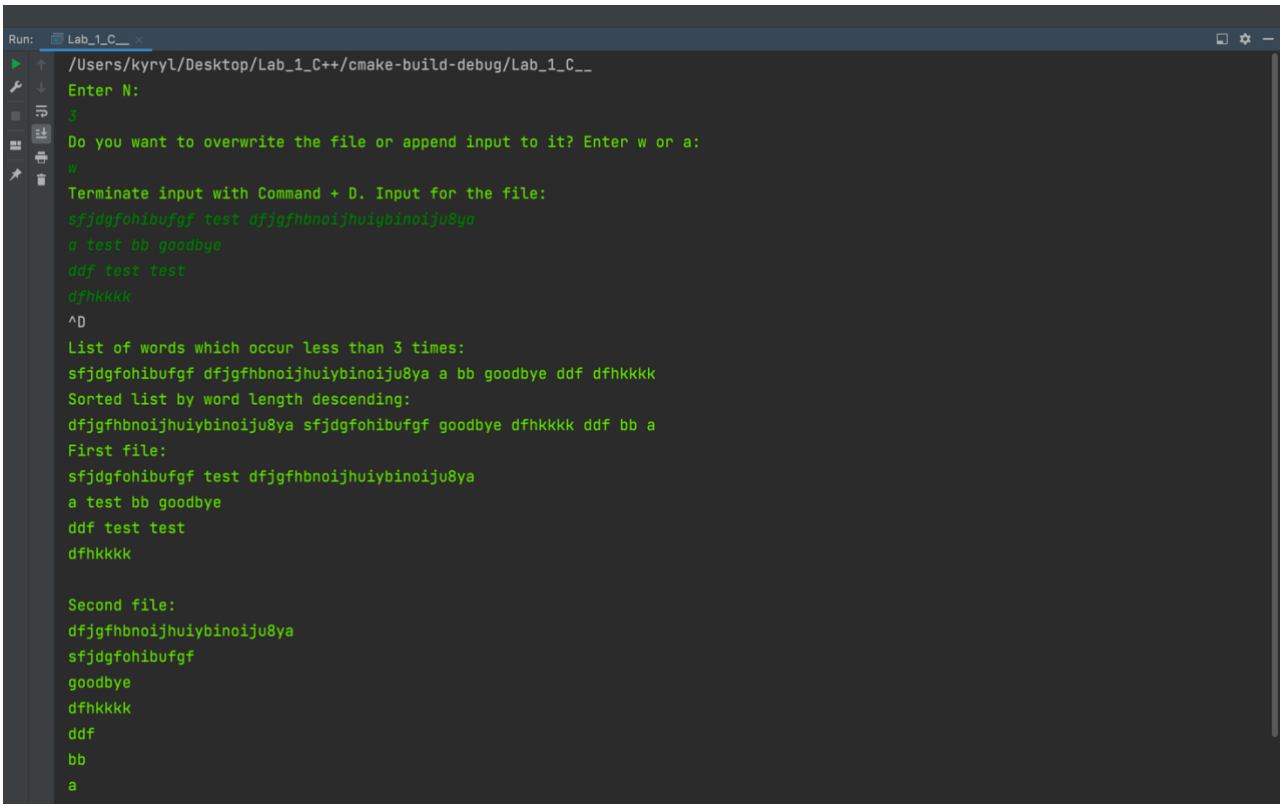
file_operations.py

```
def enter_file_mode():
    file_mode = input('Do you want to overwrite the file or
append input to it? Enter w or a: ')
    while file_mode != 'w' and file_mode != 'a':
        print("Incorrect input. Enter 'w' or 'a'.")
        file_mode = input('Do you want to overwrite the file
or append input to it? Enter w or a: ')
    return file_mode


def create_first_file(file_name, mode):
```

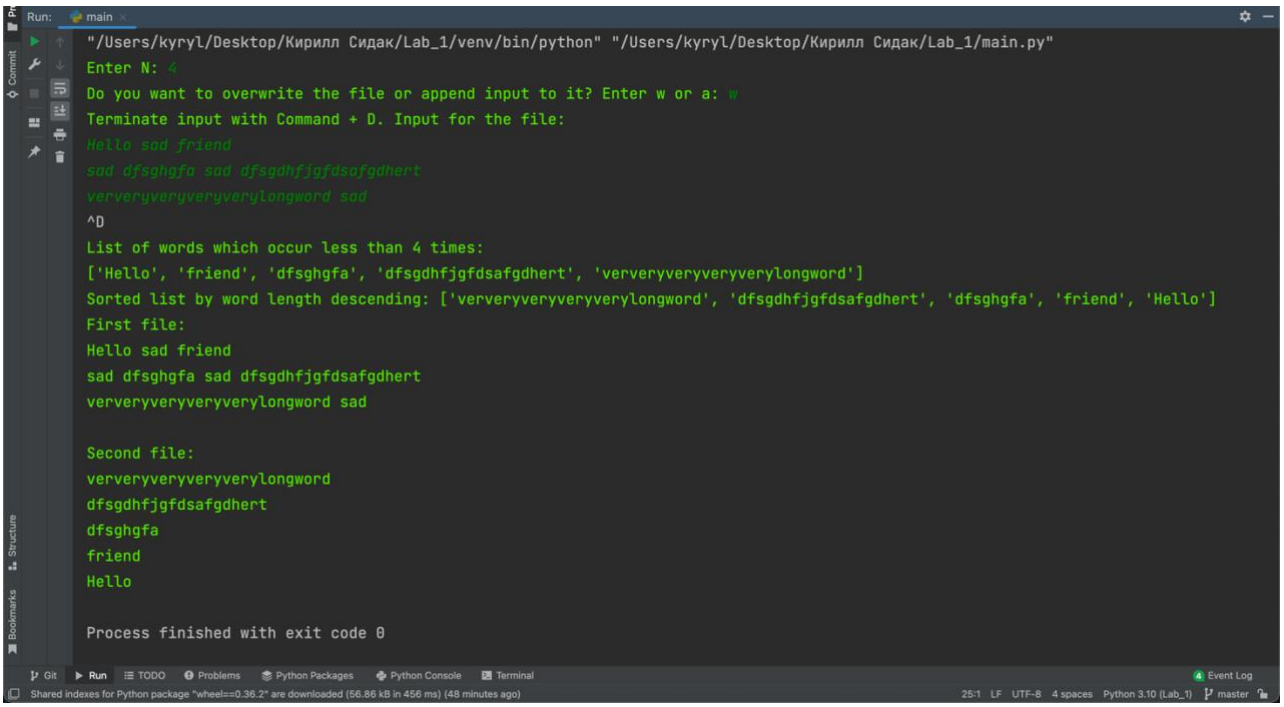
```
    with open(file_name, mode) as file:
        is_end = False
        while not is_end:
            try:
                text = input()
                if len(text) > 0:
                    file.write(text + '\n')
            except EOFError:
                is_end = True


def get_words_list(file_name, n):
    with open(file_name) as file:
        words_list = []
        for line in file:
            words = line.split()
            for word in words:
                words_list.append(word)
    for word in words_list:
        count = words_list.count(word)
        if count > 1:
            count = count if count >= n else count - 1
            for _ in range(count):
                words_list.remove(word)
    print(f'List of words which occur less than {n}
times:\n{words_list}')
    words_list.sort(key=lambda string: len(string),
reverse=True)
    return words_list


def create_second_file(file_name, words_list):
    with open(file_name, 'w') as file:
        length = len(words_list)
        for i in range(length):
            if i != length - 1:
                file.write(words_list[i] + '\n')
            else:
                file.write(words_list[i])


def output_file(file_name):
    with open(file_name) as file:
        text = file.read()
        print(text)
```

Скріншоти виконання:

C++


Run: Lab_1_C__
/Users/kyryl/Desktop/Lab_1_C++/cmake-build-debug/Lab_1_C__
Enter N:
3
Do you want to overwrite the file or append input to it? Enter w or a:
w
Terminate input with Command + D. Input for the file:
sfjdgfohibufgf test dfjgfhbnoijhuiybinoiju8ya
a test bb goodbye
ddf test test
dfhkkkk
^D
List of words which occur less than 3 times:
sfjdgfohibufgf dfjgfhbnoijhuiybinoiju8ya a bb goodbye ddf dfhkkkk
Sorted list by word length descending:
dfjgfhbnoijhuiybinoiju8ya sfjdgfohibufgf goodbye dfhkkkk ddf bb a
First file:
sfjdgfohibufgf test dfjgfhbnoijhuiybinoiju8ya
a test bb goodbye
ddf test test
dfhkkkk
Second file:
dfjgfhbnoijhuiybinoiju8ya
sfjdgfohibufgf
goodbye
dfhkkkk
ddf
bb
a

Python


Run: main
"/Users/kyryl/Desktop/Кирилл Сидак/Lab_1/venv/bin/python" "/Users/kyryl/Desktop/Кирилл Сидак/Lab_1/main.py"
Enter N: 4
Do you want to overwrite the file or append input to it? Enter w or a: w
Terminate input with Command + D. Input for the file:
Hello sad friend
sad dfsghgfa sad dfsgdhfjgfdsafgdhert
ververyveryveryverylongword sad
^D
List of words which occur less than 4 times:
['Hello', 'friend', 'dfsghgfa', 'dfsgdhfjgfdsafgdhert', 'ververyveryveryverylongword']
Sorted list by word length descending: ['ververyveryveryverylongword', 'dfsgdhfjgfdsafgdhert', 'dfsghgfa', 'friend', 'Hello']
First file:
Hello sad friend
sad dfsghgfa sad dfsgdhfjgfdsafgdhert
ververyveryveryverylongword sad
Second file:
ververyveryveryverylongword
dfsgdhfjgfdsafgdhert
dfsghgfa
friend
Hello
Process finished with exit code 0
Git
Run
TODO
Problems
Python Packages
Python Console
Terminal
Event Log
Shared indexes for Python package "wheel==0.36.2" are downloaded (56.86 kB in 456 ms) (48 minutes ago)
25:1 LF UTF-8 4 spaces Python 3.10 (Lab_1) master

Висновок

Отже, вивчити особливості створення і обробки текстових файлів даних, а саме створив два файли та реалізував операції читання з файлу та введення у файл. Зчитавши текст з одного текстового файлу, я створив список зі слів цього файлу, відсортував його за спаданням довжини слів. Записавши елементи цього списку в новий файл, я отримав коректний результат.