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. Модульне програмування»

«Текстові файли» Варіант <u>3</u>

Виконав студент <u>ІП-15, Борисик Владислав Тарасович</u>

(шифр, прізвище, ім'я, по батькові)

Лабораторна робота №1 Текстові файли Варіант <u>3</u> Задача

3. Створити текстовий файл. Слова у файлі відділені пробілами, комами, крапками. У кожному реченні визначити саме довге слово. Створити новий текстовий файл, у якому кожне речення розміщується у окремому рядку і починається із числа, що дорівнює довжині самого довгого слова в ньому, за яким йде саме це слово. Вивести вміст вихідного і створеного файлів.

Код

C++

main.cpp:

```
#include "functions.h"
using namespace std;
int main() {
    // читаємо текст з консоли, поки не буде натиснуто комбінацію Ctrl+D
    string text_to_write = capture_text();
    // записуємо цей текст у файл input.txt
    write empty file("input.txt", text_to_write);
    // читаємо текст з файлу input.txt у змінну text
    string text = read_input_file("input.txt");
    printf("\nText on input:\n%s\n\n", text.c_str());
    // вектор з реченнями тексту
    vector<string> sentences = split_sentences(text);
    // виводимо елементи вектора в консоль
    print vector(sentences);
    // вектор з найдовшими словами кожного реченя
    vector<string> longest words = get longest words(sentences);
    // виводимо найдовші слова в консоль
    print_longest_words(longest_words);
    // записуємо текст у файл output.txt
    write_file("output.txt", longest_words, sentences);
    // читаємо записаний текст з файлу output.txt
    read output file("output.txt");
}
```

functions.h:

```
#ifndef INC_2LABWORK_1_FUNCTIONS_H
#define INC_2LABWORK_1_FUNCTIONS_H
#endif
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
/** Отримує текст з клавітури, поки не натиснута комбінація клавіш Ctrl+Enter
* @return: текст з клавітури
*/
string capture_text();
/** Виводить елементі вектора в консоль
* @param vec: вектор
*/
void print_vector(const vector<string>& vec);
/** Читає текст з вхідного файлу
* @param file_name: назва файлу
 * @return: текст з файлу
*/
string read_input_file(const string& file_name);
/** Ділить речення у тексті у вектор
* @param text: текст
* @return: вектор з реченнями
 */
vector<string> split sentences(const string& text);
/** Виводить в консоль найдовші слова і їй довжину
* @param longest_words: вектор
*/
void print_longest_words(const vector<string>& longest_words);
/** Отримує найдовші слова із вектора речень
* @param sentences: вектор з реченнями
 * @return: вектор з найдовшими словами
 */
vector<string> get longest words(const vector<string>& sentences);
/** Записує текст у файл
* @param file_name: найзва файлу
* @param longest_words: вектор з найдовшими словами речень
   @param sentences: вектор з реченнями
*/
void write_file(const string& file_name, const vector<string>& longest_words,
vector<string>& sentences);
/** Записує текст у файл input
 * @param file_name: найзва файлу
```

```
* @param text: текст
*/
void write_empty_file(const string& file_name, const string& text);
/** Читає текст з вихідного файлу
* @param file_name: назва файлу
*/
void read_output_file(const string& file_name);
```

functions.cpp:

```
#include "functions.h"
string capture_text(){
    cout << "Enter your text\nPress Enter to go to the next line\nPress Ctrl +</pre>
D - to end writing\n";
    string text;
    string line;
    // ascii код комбінації клавіш Ctrl + D
    int ascii code = 94;
    // поки не натиснута комбінація клавіш Ctrl + D
    while(int(line[0]) != ascii code){
        getline(cin, line);
        text += line + '\n';
    }
    // відкидаємо 4 зайвих символи (2 зайвих \n i ^D)
    text = text.substr(0, text.size()-4);
    return text;
}
void write_empty_file(const string& file_name, const string& text){
    // потік write file (для читання і запису)
    ofstream write_file(file_name);
    // записуємо текст у файл
    write file << text;
    // закриваємо потік(файл)
    write_file.close();
}
void print vector(const vector<string>& vec){
    cout << "Sentences vector:\n";</pre>
    for(auto & i : vec){
        cout << i << "\n";</pre>
    cout << "\n";</pre>
}
string read input file(const string& file name){
    // потік read file (тільки для читання)
    ifstream read file(file name);
    string text;
    string line;
    // поміщаємо текст з потоку у змінну line
    while (getline(read file, line)){
        text += line + '\n';
```

```
// закриваємо потік(файл)
    read file.close();
    // відкидуємо зайвий символ
    text = text.substr(0, text.size()-1);
   return text;
}
vector<string> split_sentences(const string& text){
    // вектор з реченнями
    vector<string> sentences;
    // позиція минулої крапки
    int previous position = 0;
    // лічильник (рахує символи тексту)
    int counter = 0;
   while(counter <= text.size()){</pre>
        // якщо символ дорівнює крапці або (counter дорівнює останньому символу
тексту і минулий символ не дорівнює крапці)
        // (counter == text.size() && text[counter-1] != '.') потрібно для
того, що б цикл корректно виділив останнє речення, якщо воно записане без
крапки
        if(text[counter] == '.' || text[counter] == '\n' || (counter ==
text.size() && text[counter-1] != '.')){
            // речення (підрядок, який починається з позиції минулої крапки
довжиною counter-previous position)
            string sentence = text.substr(previous_position, counter-
previous position);
            // добавляємо це речення у вектор
            sentences.push back(sentence);
            // оновлюємо позицію крапки
            previous_position = counter+1;
        }
        counter++;
    }
   return sentences;
}
vector<string> split words(const vector<string>& sentences, int i){
    // вектор зі словами
    vector<string> words;
    // позиція минулої крапки
    int previous position = 0;
    // лічильник (рахує символи тексту)
    int counter = 0;
    while(counter <= sentences[i].size()){</pre>
        // якщо символ дорівнює розділюнику або останньому символу слова
        if(sentences[i][counter] == ',' || sentences[i][counter] == ' ' ||
```

```
counter == sentences[i].size()){
            // слово (підрядок, який починається з позиції минулого розділювача
довжиною counter-previous position)
            string word = sentences[i].substr(previous position, counter-
previous position);
            // добавляємо це слово у вектор
            words.push back(word);
            // оновлюємо позицію
            previous position = counter+1;
        counter++;
    }
   return words;
}
void print longest words(const vector<string>& longest words){
    for(int i =0; i < longest words.size(); i++){</pre>
        printf("Longest word in sentence #%d is: \"%s\" and its length: %d\n",
i+1, longest words[i].c str(), longest words[i].length());
    cout << "\n";
}
vector<string> get_longest_words(const vector<string>& sentences){
    // вектор з найдовшими словами
    vector<string> longest_words;
    for(int i=0; i < sentences.size(); i++){</pre>
        // ділимо речення на слова
        vector<string> words = split words(sentences, i);
        // найдовше слово (перше слово у реченні)
        string longest word = words[0];
        for(int j=1; j<words.size();j++){</pre>
            // якщо довжина слова більша за довжину longest word
            if(words[j].size() > longest word.size()){
                // оновлюємо найдовше слово
                longest word = words[j];
        }
        // добавляємо найдовше слово у вектор
        longest words.push back(longest word);
    }
    return longest_words;
}
void write_file(const string& file_name, const vector<string>& longest_words,
vector<string>& sentences){
```

```
// потік write_file (для читання і запису)
    ofstream write_file(file_name);
    // записуємо текст у файл
    for(int i=0; i < sentences.size(); i++){</pre>
        write file << to string(longest words[i].size()) + " " +</pre>
longest_words[i] + " | " + sentences[i] + "\n";
    // закриваємо потік(файл)
    write_file.close();
}
void read_output_file(const string& file_name){
    // потік read_output_file (тільки для читання)
    ifstream read_output_file(file_name);
    // виводимо в консоль текст з файлу
    cout << read_output_file.rdbuf() << "\n";</pre>
    // закриваємо потік(файл)
    read output file.close();
}
```

Python

main.py:

```
from functions import *
# читаємо текст з консолі, поки не буде натиснуто клавішу Ctrl + D
text_to_write = capture_text()
# записуємо цей текст у файл input.txt
write_empty_file("input.txt", text_to_write)
# читаємо текст з файлу input.txt у змінну text
text = read file("input.txt")
print(f"Text on input:\n{text}\n")
# список з реченнями тексту
sentences = split_sentences(text)
print("Sentences list:")
print(*sentences, sep='\n')
# вектор з найдовшими словами кожного реченя
longest_words = get_longest_words(sentences)
print_longest_words(longest_words)
# записуємо текст у файл output.txt
write_to_file("output.txt", longest_words, sentences)
output text = read file("output.txt")
print(output_text)
```

functions.py:

```
def capture_text():
    Читає текст з консолі, поки не буде натиснуто клавішу Еsc
    :return: текст з консолі
    print('Enter your text\nPress Enter to go to the next line\nPress Ctrl + D - to
end writing')
    import sys
    text = sys.stdin.read()
    # відкидаємо зайвий символ \n
    text = text[:len(text) - 1]
    print(f"Whole text:\n{text}\n")
    return text
def write empty file(file name: str, text: str):
    Записуємо текст у файл
    :param file_name: назва файлу
    :param text:
                    текст
    write_file = open(file_name, 'wt')
    write file.write(text)
    write_file.close()
def get_longest_words(sentences: list[str]):
    Отримує найдовші слова із списку речень
    :param sentences: список з реченнями
                      список з найдовшими словами
    :return:
    # список з найдовшими словами
    longest words = []
    for i in range(len(sentences)):
        # заміняємо у реченні коми на пробіли (потрібно для того, щоб корректно
відділити слова)
       new_sentence = sentences[i].replace(',', '')
        # список із словами
        words = new sentence.split(' ')
        # найдовше слово (перше слово у реченні)
        longest_word = words[0]
        for j in range(1, len(words)):
            # якщо довжина слова більша за довжину longest word
            if len(words[j]) > len(longest_word):
                longest word = words[j]
```

```
# добавляємо найдовше слово у вектор
        longest words.append(longest word)
    return longest words
def print longest words(longest words: list[str]):
    Виводить в консоль найдовші слова і їй довжину
    :param Longest_words: список із словами
    print('\n')
    for i in range(len(longest_words)):
       print(f'Longest word in sentence #{i+1} is: "{longest_words[i]}" and its
length: {len(longest_words[i])}')
    print('\n')
def split sentences(text: str):
   Ділить речення у тексті
    :param text: mekcm
    :return: список з реченнями
    text = text.replace('\n', '.')
    sentences = text.split('.')
    for i in range(len(sentences)):
        # перевіряєємо, чи є у списку "пусті речення" (потрібно для того, щоб у список
не заносився пустий елемент)
        if sentences[i] == '':
            sentences.pop(i)
    return sentences
def write_to_file(file_name: str, longest_words: list[str], sentences: list[str]):
    Записує текст у файл
    :param file_name:
                      найзва файлу
    :param longest_words: список з найдовшими словами речень
    :param sentences: список з реченнями
    :return:
    write_file = open(file_name, 'wt')
    for i in range(len(longest words)):
        write_file.write(f"{len(longest_words[i])} {longest_words[i]} |
{sentences[i]}\n")
    write file.close()
def read_file(file_name: str):
```

```
.....
```

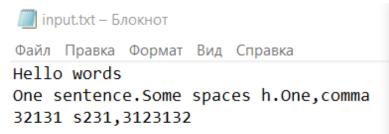
return text

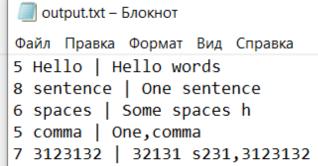
Результат виконання програми

C++:

```
Enter your text
Press Enter to go to the next line
Press Ctrl + D - to end writing
^D
Text on input:
Hello words
One sentence. Some spaces h. One, comma
32131 s231,3123132
Sentences vector:
Hello words
One sentence
Some spaces h
One, comma
32131 s231,3123132
Longest word in sentence #1 is: "Hello" and its length: 5
Longest word in sentence #2 is: "sentence" and its length: 8
Longest word in sentence #3 is: "spaces" and its length: 6
Longest word in sentence #4 is: "comma" and its length: 5
Longest word in sentence #5 is: "3123132" and its length: 7
```

```
5 Hello | Hello words
8 sentence | One sentence
6 spaces | Some spaces h
5 comma | One,comma
7 3123132 | 32131 s231,3123132
```





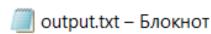
Python:

```
Enter your text
Press Enter to go to the next line
Press Ctrl + D - to end writing
^D
Whole text:
Hello words
Another sentence.One space.And, some, commas
32131 s231,3123132
Text on input:
Hello words
Another sentence.One space.And, some, commas
32131 s231,3123132
Sentences list:
Hello words
Another sentence
One space
And, some, commas
32131 s231,3123132
```

```
Longest word in sentence #1 is: "Hello" and its length: 5
Longest word in sentence #2 is: "sentence" and its length: 8
Longest word in sentence #3 is: "space" and its length: 5
Longest word in sentence #4 is: "commas" and its length: 6
Longest word in sentence #5 is: "3123132" and its length: 7
5 Hello | Hello words
8 sentence | Another sentence
5 space | One space
6 commas | And, some, commas
7 3123132 | 32131 s231,3123132
```



Файл Правка Формат Вид Справ файл Правка Формат Вид Справка Hello words 32131 s231,3123132



5 Hello | Hello words Another sentence.One space 8 sentence | Another sentence 5 space | One space 6 commas | And, some, commas 7 3123132 | 32131 s231,3123132