# 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»

Варіант 13

Виконав студент Котков Тимур Максимович

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

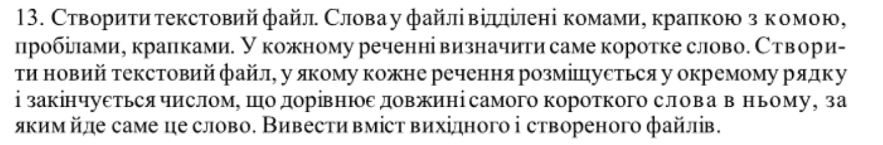
Київ 2022

**Лабораторна робота №1**

**Файли даних**

**Мета –** вивчити особливості створення і обробки текстових файлів даних.

**Індивідуальне завдання:**



**Постановка задачі**

Створимо текстовий файл, відкоригуємо та напишемо речення з окремого рядка. Розділимо речення на слова, знайдемо найменші слова в кожному реченні. Створимо новий текстовий файл.

**Код на С++ :**

**Main:**

#include<iostream>

#include <fstream>

#include <vector>

#include"Header.h"

using namespace std;

int main()

{

string file1\_p = "file\_1.txt", file2\_p = "file\_2.txt";

ofstream file1 = createFile(file1\_p);

vector<string> text1 = inputText(file1\_p);

fillFile(file1\_p, text1);

vector<vector<string>> splitedText = split(file1\_p);

vector<string> minWords;

vector<int> minLength = findMinWord(splitedText, minWords);

ofstream file2 = createFile(file2\_p);

vector<string> text2 = createText(splitedText, minWords, minLength);

fillFile(file2\_p, text2);

cout << "\nFile after changes: " << endl;

outputFile(file2\_p);

}

**Source:**

#include <string.h>

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include"Header.h"

using namespace std;

ofstream createFile(string file\_name) {

ofstream file;

file.open(file\_name, ios::out | ios::trunc);

if (!file.is\_open()) {

cout << "File isn't created" << endl;

}

else {}

file.close();

return file;

}

void fillFile(string file\_name, vector<string> inputText) {

ofstream file;

file.open(file\_name);

for (int i = 0; i < inputText.size(); i++)

{

file << inputText[i];

file << "\n";

}

file.close();

}

vector<string> inputText(string file\_name) {

vector<string> input;

string str;

cout << "Input text for " + file\_name + ": " << endl;

while (getline(cin, str)) {

input.push\_back(str);

}

return input;

}

vector<string> createText(vector<vector<string>> splitedText, vector<string> minWords, vector<int> minLength) {

vector<string> text;

for (int i = 0; i < splitedText.size() - 1; i++)

{

string str;

for (int j = 0; j < splitedText[i].size(); j++)

{

str += splitedText[i][j] + " ";

}

str += "(" + minWords[i] + "-";

str += to\_string(minLength[i]) + ")";

text.push\_back(str);

}

return text;

}

vector<string> extractText(string file\_name) {

vector<string> text;

string str;

ifstream file;

file.open(file\_name, ios::in);

if (!file.is\_open()) {

cout << "Cannot open file!" << endl;

}

else {}

while (!file.eof()) {

getline(file, str);

text.push\_back(str);

}

file.close();

return text;

}

vector<vector<string>> split(string file\_name) {

vector<vector<string>> splitedText;

vector<string> text = extractText(file\_name);

for (int i = 0; i < text.size(); i++)

{

vector<string> words;

string word, sti = text[i] + " ";

for (int j = 0; j < sti.size(); j++) {

if (!ispunct(sti[j]) && !isblank(sti[j])) {

word += sti[j];

}

else {

if (word != "") {

words.push\_back(word);

word = "";

}

else {

continue;

}

}

}

splitedText.push\_back(words);

}

return splitedText;

}

vector<int> findMinWord(vector<vector<string>> words, vector<string>& minWords) {

vector<int> minLength;

for (int i = 0; i < words.size() - 1; i++)

{

string min\_word = words[i][0];

int min\_length = words[i][0].size();

for (int j = 0; j < words[i].size(); j++) {

if (min\_word.size() >= words[i][j].size()) {

min\_word = words[i][j];

min\_length = words[i][j].size();

}

}

minWords.push\_back(min\_word);

minLength.push\_back(min\_length);

}

return minLength;

}

void outputFile(string file\_name) {

ifstream file;

string str;

file.open(file\_name);

while (getline(file, str)) {

cout << str << endl;;

}

file.close();

}

**Header:**

#pragma once

using namespace std;

ofstream createFile(string file\_name);

void fillFile(string file\_name, vector<string> inputText);

vector<vector<string>> split(string file\_name);

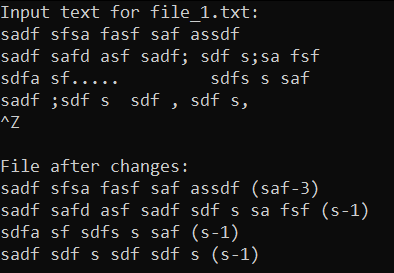
vector<string> extractText(string file\_name);

vector<int> findMinWord(vector<vector<string>> words, vector<string> &minWords);

vector<string> inputText(string file\_name);

vector<string> createText(vector<vector<string>> splitedText, vector<string> minWords, vector<int> minLength);

void outputFile(string file\_name);



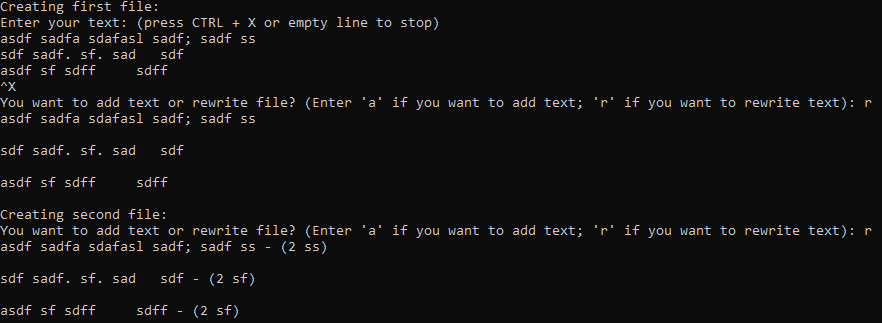
**Результати у С++:**

**}**

**Код на Python :**

*import* re  
  
  
*def* input\_text():  
 text = []  
 print("Enter your text: (press CTRL + X or empty line to stop)")  
 flag = *True* exit\_line = "\u0018"  
 *while* flag:  
 line = str(input())  
 *if* (line == exit\_line) *or* (line == ""):  
 flag = *False  
 else*:  
 text.append(line)  
 *return* text  
  
  
*def* create\_file(file\_name: str, text=*None*, enc='utf-8'):  
 flag = *True  
 while* flag:  
 choice = input("You want to add text or rewrite file? "  
 "(Enter 'a' if you want to add text; 'r' if you want to rewrite text): ")  
 *if* choice == 'a':  
 new\_file = open(file\_name, 'a', encoding=enc)  
 new\_file.write('\n' + text[0])  
 flag = *False  
 if* choice == 'r':  
 new\_file = open(file\_name, 'w', encoding=enc)  
 new\_file.write(text[0])  
 flag = *False  
 else*:  
 print('Wrong symbol. Try again')  
  
 *for* i *in* range(1, len(text)):  
 new\_file.write('\n' + text[i])  
  
 new\_file.close()  
  
  
*def* find\_min\_word(text):  
 min\_words = []  
 *for* i *in* range(len(text)):  
 buff = re.findall(r"[\w']+", text[i])  
 min\_words.append(min((word *for* word *in* buff *if* word), key=len))  
  
 *return* min\_words  
  
  
*def* print\_file(file):  
 lines = (open(file, 'r').readlines())  
 *for* string *in* lines:  
 print(string)  
  
  
*if* \_\_name\_\_ == '\_\_main\_\_':  
 print('Creating first file: ')  
 text = input\_text()  
 create\_file('first\_file.txt', text)  
 print\_file('first\_file.txt')  
 shortest\_words = find\_min\_word(text)  
  
 print('\nCreating second file: ')  
 lines = []  
 *for* i *in* range(len(text)):  
 lines.append(text[i] + f' - ({len(shortest\_words[i])} {shortest\_words[i]})')  
 create\_file('second\_file.txt', lines)  
 print\_file('second\_file.txt')

**Скріншот результатів програми на Python :**

****

**Висновок:**

Під час лабораторної роботи ми вивчили особливості створення і обробки текстових файлів даних на двох мовах програмування: C++ та Python. Оскільки програма правильно створює та виводить текстові файли, знаходить мінімальні слова в реченнях та має вірний вивід, то вона працює правильно.