

# ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ № 9

За курсом «Програмування»

Студента групи ПА-23-1

Мороза Миколи Олександровича

Кафедра комп'ютерних технологій, ДНУ

2023/2024

## 1. Постановка задачі

Варіант 12.

### 1. Обробка текстової інформації, що міститься у масиві символів

Завдання: Виконати задані дії із введенним текстом.

Індивідуальне завдання №1

12. Знайти саме довге слово в реченні та надрукувати його в зверненому вигляді.

### 2. Рядкові функції.

Завдання: Написати програму, в якій були б розроблені 2 функції, що імітують дії стандартних рядкових функцій (див. індивідуальне завдання). Вибір функцій здійснюється за допомогою пунктів меню.

Індивідуальне завдання

Варіант №12

Функції: `strpbrk( )` та `strtoul( )`

### 3. Обробка текстової інформації, що міститься у файлах

1. За допомогою текстового редактора створити файл, що містить текст. Довжина рядка тексту не повинна перевищувати 80 символів. Це вхідний файл.

2.Програма повинна 1)інформацію вхідного файлу записати у вихідний файл; 2)реалізувати дії, вказані в індивідуальному завданні і занести відповідну інформацію у вихідний файл.

3.Ім'я вхідного файлу задає користувач в командному рядку або під час виконання програми.

4.Ім'я вихідного файлу: перші символи -не більше трьох символів з імені вхідного файлу, а решти символів \_out.

5.Вихідний файл повинен мати розширення .dat.

6.Програма повинна забезпечувати по запиту користувача: 1) виведення тексту вхідного файлу на екран дисплея; 2) виведення тексту вихідного файлу на екран дисплея.

7.Закінчені послідовності дій оформити у вигляді функцій. Всі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

8.Для вводу і виводу інформації (і для консолі, і для текстових файлів) застосувати засоби форматowanego вводу/виводу.

## Варіант №12

Визначити кількість рядків, що складаються із заданої користувачем кількості слів. Вивести ці рядки. 9.Визначення функцій і головну функцію розмістити в двох окремих файлах. Індивідуальне завдання

## 2.Опис розв'язку

По-перше для правильності виконання ми підключимо бібліотеки `iostream`, `Windows.h`, `fstream`, `cstring` та пропишемо `using namespace std`, та пропишемо директиву перепроцесора `_CRT_SECURE_NO_WARNINGS`.

У першому завданні ми просто отримуємо рядок інформації від користувача, далі ми розбиваємо цей рядок на лексеми за допомогою `strtok`, після того за допомогою `strlen` порівнюємо наші слова, знаходимо найбіль та записуємо його в змінну, далі просто виводимо це слово у зворотному порядку

У другому завданні першій підзадачі для функції `strpbrk`, у двовимірному циклі порівнюємо кожен елемент нашого речення з кожним елементом

елементів для перевірки на перше входження, якщо є перше сходження то повертаємо його, якщо немає, то повертаємо нульовий показчик.

Для другої підзадачі, функції `strtol`, ми першим ділом перевіряємо чи є у рядку пробіли, далі ми визначаємось із знаком числа, після того визначаємось із в якій системі числення написано число, після всіх цих дій ми робимо цикл який перетворює рядок в число.

Для третього завдання, ми отримувемо порядково інформацію з файлу, але максимальна кількість символів буде 80, далі ми розбиваємо цей рядок на лексеми, після цього рахуємо кількість цих слів у реченні, якщо кількість співпадає то виводимо зміну у яку попередньо скопіювали наше речення

### 3. Вихідний текст програми розв'язку задачі

Файл: `project.cpp`

```
#include "project.h"
#include <Windows.h>
#include <fstream>
#include <iostream>
#include <cstring>

using namespace std;

int my_strlen(const char* str) {
    int length = 0;
    for (int i = 0; str[i] != '\0'; i++) {
        ++length;
    }
    return length;
}

char* my_strtok(char* str, const char* dels) {
    static char* nextToken = nullptr;

    if (str != nullptr) {
        nextToken = str;
    }

    if (nextToken == nullptr) {
        return nullptr;
    }

    char* start = nextToken;
    while (*start) {
        bool isDelimiter = false;
        for (const char* d = dels; *d != '\0'; ++d) {
            if (*start == *d) {
                isDelimiter = true;
                break;
            }
        }
    }
```

```

        }
        if (!isDelimiter) {
            break;
        }
        ++start;
    }

    if (*start == '\0') {
        nextToken = nullptr;
        return nullptr;
    }

    char* end = start;
    while (*end) {
        bool isDelimiter = false;
        for (const char* d = dels; *d != '\0'; ++d) {
            if (*end == *d) {
                isDelimiter = true;
                break;
            }
        }
        if (isDelimiter) {
            break;
        }
        ++end;
    }

    if (*end != '\0') {
        *end = '\0';
        nextToken = end + 1;
    }
    else {
        nextToken = nullptr;
    }

    return start;
}

```

```

void algorithm_for_first_task(char* word, char*& longest_word, int& longest_word_len, char* delim) {
    while (word) {
        if (my_strlen(word) > longest_word_len) {
            longest_word = word;
            longest_word_len = my_strlen(word);
        }
        word = my_strtok(NULL, delim);
    }
}

```

```

char* func_strpbrk(char* sentence, char* elements) {
    while (*sentence != '\0') {
        for (char* c = elements; *c != '\0'; ++c) {
            if (*sentence == *c) {
                return sentence;
            }
        }
        ++sentence;
    }
}

```

```

long func_tol(const char* str, char** endptr, int base) {
    const char* s = str;
    long result = 0;
    int sign = 1;

    // Пропускаємо початкові пробіли
    while (isspace(*s)) {
        s++;
    }
    // Визначення знака числа
    if (*s == '-') {
        sign = -1;
        s++;
    }
    else if (*s == '+') {
        s++;
    }
    // Визначення основи числа
    if (base == 0) {
        if (*s == '0') {
            if (s[1] == 'x' || s[1] == 'X') {
                base = 16;
                s += 2;
            }
            else {
                base = 8;
                s++;
            }
        }
        else {
            base = 10;
        }
    }
    else if (base == 16) {
        if (s[0] == '0' && (s[1] == 'x' || s[1] == 'X')) {
            s += 2;
        }
    }

    // Основний цикл перетворення рядка в число
    while (*s) {
        int digit = 0;

        if (isdigit(*s)) {
            digit = *s - '0';
        }
        else if (*s >= 'a' && *s <= 'z') {
            digit = *s - 'a' + 10;
        }
        else if (*s >= 'A' && *s <= 'Z') {
            digit = *s - 'A' + 10;
        }
        else {
            break;
        }

        if (digit >= base) {
            break;
        }

        // Перевірка на переповнення
        if (result > (LONG_MAX - digit) / base) {

```

```

        result = (sign == 1) ? LONG_MAX : LONG_MIN;
        if (endptr != nullptr) {
            *endptr = (char*)s;
        }
        return result;
    }

    result = result * base + digit;
    s++;
}

if (endptr != nullptr) {
    *endptr = (char*)s;
}

return result * sign;
}

```

```

void Third_task_fstream(int count_of_words, const char* input, const char* output) {
    ifstream fin(input);
    ofstream fout(output, ios::app);
    if (!fin.is_open()) {
        fout << "Файл " << input << " не було відкрито за певних умов" << endl;
    }
    if (!fout.is_open()) {
        fout << "Файл " << output << " не було відкрито за певних умов" << endl;
    }

    fout << endl << "Наші речення: " << endl;

    char output_word_fille_1[81];
    while (fin.getline(output_word_fille_1, 80)) {
        fout << output_word_fille_1 << endl;
    }

    fout.close();
    fin.close();
    fout.open(output, ios::app);
    fin.open(input);

    char* word_task_third;
    char sentence_task_third[81];
    char prototype_sentence[81];
    int count_of_sentence = 0;
    int count_task_third = 0;
    cout << endl << endl;
    fout << endl << "Результат дії: " << endl;
    while (fin.getline(sentence_task_third, 80)) {

        strcpy(prototype_sentence, sentence_task_third);
        word_task_third = strtok(sentence_task_third, " ");
        while (word_task_third) {
            count_task_third++;
            word_task_third = strtok(NULL, ".,/\\\"';:[]{}()~*! ?");
        }

        if (count_task_third == count_of_words) {
            fout << prototype_sentence << endl;
            count_of_sentence++;
        }
    }
}

```

```

    }

    count_task_third = 0;
}
fout << "Всього " << count_of_sentence << " речень з " << count_of_words << " словами " << endl << endl;

fout.close();
fin.close();

}}

```

## Файл: Лабораторна робота №9\_2.cpp

```

#include <iostream>
#include <Windows.h>
#include <fstream>
#include <cstring>
#include "project.h"

using namespace std;

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    cout << "\t\tЛабораторна робота №9\n\n";

    cout << "\n\t\tЗавдання №1\n\n";

    char phrase[101];
    cout << "Введіть речення(але максимальна кількість символів буде 100): ";
    cin.getline(phrase, 100);
    char* word;
    char* longest_word = nullptr;
    int longest_word_len = 0;
    char delim[] = ".,/\\\"';:[]{}()~*!?!? ";
    word = my_strtok(phrase, " ");

    algoritm_for_first_task(word, longest_word, longest_word_len, delim);

    if (longest_word_len) {
        cout << "Розмір найдовшого слова: " << longest_word_len << endl;
        cout << "Саме слово: ";
        for (int i = longest_word_len; i >= 0; i--) {
            cout << longest_word[i];
        }
        cout << endl;
    }
    else {
        cout << "Ви не ввели жодного слова" << endl;
    }

    cout << "\n\n\t\tЗавдання №2\n\n";

    int choose_an_option_for_task_2;

```

```

char sentence[100], elements[10];

while (true) {
    cout << endl << "Введіть число для вибору функції\n";
    cout << "Де 0 - ви нічого не обираєте, та йдете на наступне завдання\n";
    cout << "1 - обираєте функцію strpbrk, 2 - обираєте функцію strtol\n";
    cout << "Введіть цифру: ";
    cin >> choose_an_option_for_task_2;
    while (!cin) {
        cin.clear();
        cin.ignore(32767, '\n');
        cout << "Ви ввели не число" << endl;
        cout << "Введіть цифру: ";
        cin >> choose_an_option_for_task_2;
    }
    if (choose_an_option_for_task_2 == 1) {
        cin.ignore(32767, '\n');

        cout << "Введіть речення, але максимальна довжина його 100 символів: ";
        cin.getline(sentence, sizeof(sentence));
        cout << "Введіть символи для пошуку, але максимальна кількість 10: ";
        cin.getline(elements, sizeof(elements));

        char* ptr_1 = func_strpbrk(sentence, elements);
        cout << "Робота прописаної функції" << endl;
        if (ptr_1 != nullptr) {
            cout << "Перше входження: " << *ptr_1 << endl;
        }
        else {
            cout << "Жодних відповідей не знайдено." << endl;
        }
        cout << "Робота стандартної функції" << endl;
        char* ptr_2 = strpbrk(sentence, elements);

        if (ptr_2 != nullptr) {
            cout << "Перше входження: " << *ptr_2 << endl;
        }
        else {
            cout << "Жодних відповідей не знайдено." << endl;
        }
    }
    else if (choose_an_option_for_task_2 == 2) {
        char number_sentence[101];
        int type_num;
        cin.ignore(32767, '\n');
        cout << "Введіть якесь число: ";
        cin.getline(number_sentence, 100);
        cout << "Введіть в якій із систем числення буде повертати \nУ разі некоректного введення  
то поверне у 10 річній системі" << endl;
        cout << "Введіть систему числення: ";
        cin >> type_num;
        char* end;
        long value = func_tol(number_sentence, &end, type_num);
        cout << "Результат роботи прописаної функції: " << value << endl;
        long int number_2 = strtol(number_sentence, &end, type_num);
        cout << "Результат роботи стандартної функції: " << number_2 << endl;
    }
    else {
        break;
    }
}

```



```

cout << "\tЗавдання №3\n\n";
char input[20];
char end_of_input[5] = ".txt";
cout << "Введіть ім'я вхідного файлу (тип .txt автоматично буде обраний): ";
cin.getline(input, 15);

while (strlen(input) > 15) {
    cout << "Ви ввели ім'я більше за 15 символів\n";
    cout << "Введіть ім'я файлу(тип .txt автоматично буде обраний): ";
    cin.getline(input, 10);
}
char prototype_input[15];
strcpy(prototype_input, input);
strcat(input, end_of_input);

char output[13];
char end_of_fille_name[9] = "_out.dat";

prototype_input[3] = '\0';

if (prototype_input[2] == '.') {
    prototype_input[2] = '\0';
    strcpy(output, prototype_input);
    strcat(output, end_of_fille_name);
}
else if (prototype_input[1] == '.') {
    prototype_input[1] = '\0';
    strcpy(output, prototype_input);
    strcat(output, end_of_fille_name);
}

else {
    strcpy(output, prototype_input);
    strcat(output, end_of_fille_name);
}

char temp_for_empty_file[81];
int count_of_words;
int counter_for_empty_file = 0;
bool flag = true;
cout << "Введіть кількість слів у реченні: ";
cin >> count_of_words;
while (!cin) {
    cin.clear();
    cin.ignore(32767, '\n');
    cout << "Ви ввели не число" << endl;
    cout << "Введіть кількість слів у реченні: ";
    cin >> count_of_words;
}
ifstream fin(input);
ofstream fout(output);
if (!fin.is_open()) {
    cout << "Файл " << input << " не було відкрито за певних умов" << endl;
}
else {
    cout << "Файл " << input << " було успішно відкрито" << endl;
}

```

```

if (!fout.is_open()) {
    cout << "Файл " << output << " не було відкрито за певних умов" << endl;
}
else {
    cout << "Файл " << output << " було успішно відкрито" << endl;
}
if (fin.is_open()) {
    while (fin.getline(temp_for_empty_file, 80)) {
        ++counter_for_empty_file;
    }
}
if (counter_for_empty_file == 0) {
    cout << endl << "Ваш файл пустий тому програма не може обробляти ваші дані з файлу" << endl;
    cout << "Будь ласка введіть дані у файл" << endl;
    flag = false;
}
fin.close();

if (counter_for_empty_file != 0) {
    if (flag) {
        int index_for_third_task;

        Third_task_fstream(count_of_words, input, output);

        cout << "Дані були успішно записанні у вихідний файл" << endl;
        cout << "1 - Програма виведе на екран дані з вхідного файлу" << endl;
        cout << "2 - Програма виведе на екран дані з вихідного файлу" << endl;
        cout << "будь-яке інше число, вихід з програми" << endl;
        cout << "Введіть число: ";
        cin >> index_for_third_task;
        while (true) {
            if (index_for_third_task == 1) {
                fin.open(input);
                cout << endl << "Дані з вхідного файлу: " << endl;
                char output_word_fille_1[81];
                while (fin.getline(output_word_fille_1, 80)) {
                    cout << output_word_fille_1 << endl;
                }
                fin.close();
                cout << "1 - Програма виведе на екран дані з вхідного файлу" << endl;
                cout << "2 - Програма виведе на екран дані з вихідного файлу" << endl;
                cout << "будь-яке інше число, вихід з програми" << endl;
                cout << "Введіть число: ";
                cin >> index_for_third_task;
            }
            else if (index_for_third_task == 2) {
                fin.open(output);
                cout << endl << "Дані з вихідного файлу: " << endl;
                char output_word_fille_2[81];
                while (fin.getline(output_word_fille_2, 80)) {
                    cout << output_word_fille_2 << endl;
                }
                fin.close();
                cout << "1 - Програма виведе на екран дані з вхідного файлу" << endl;
                cout << "2 - Програма виведе на екран дані з вихідного файлу" << endl;
                cout << "будь-яке інше число, вихід з програми" << endl;
                cout << "Введіть число: ";
                cin >> index_for_third_task;
            }
        }
    }
}

```

```

        }
        else {
            break;
        }
    }
}
}
fin.close();
fout.close();

system("pause");
return 0;
}

```

Файл: project.h

```
#pragma once
```

```

void algoritm_for_first_task(char* word, char*& longest_word, int& longest_word_len, char* delim);
char* func_strpbrk(char* sentence, char* elements);
long func_tol(const char* str, char** endptr, int base);
void Third_task_fstream(int count_of_words, const char* input, const char* output);
int my_strlen(const char* str);
char* my_strtok(char* str, const char* dels);

```

## 4.Опис інтерфейсу програм

Користувач запускає програму, та вона виводить щоб користувач вводив інформацію у вигляді рядків, але якщо програма попросить користувача ввести число, а користувач введе слово, то вона виведе йому помилку, у третьому завданні якщо користувач не введе нічого у вхідному файлі то програма виведе, що немає елементів у файлі.

Далі програма виведе результати її роботи та цим закінчить свою роботу, щоб вийти з програми користувачу потрібно натиснути будь-яку клавішу клавіатури.

## 5.Опис тестових прикладів

### Лабораторна робота №9

#### Завдання №1

Введіть речення(але максимальна кількість символів буде 100): dofb dohi eg eprigherg eh  
Розмір найдовшого слова: 9  
Саме слово: grehgirpe

#### Завдання №2

Введіть число для вибору функції  
Де 0 - ви нічого не обираєте, та йдете на слідуюче завдання  
1 - обираєте функцію strpbrk, 2 - обираєте функцію strtol  
Введіть цифру: 1  
Введіть речення, але максимальна довжина його 100 символів: dnbdfn hbed nrt  
Введіть символи для пошуку, але максимальна кількість 10: 5egegh  
Робота прописаної функції  
Перше вхождення: h  
Робота вбудованої функції  
Перше вхождення: h

Введіть число для вибору функції  
Де 0 - ви нічого не обираєте, та йдете на слідуюче завдання  
1 - обираєте функцію strpbrk, 2 - обираєте функцію strtol  
Введіть цифру: 2  
Введіть якесь число: 0xFF  
Введіть в якій із систем числення буде повертати  
У разі некоректного введення то поверне у 10 річній системі  
Введіть систему числення: 16  
Результат роботи прописаної функції: 255  
Результат роботи вбудованої функції: 255

Введіть число для вибору функції  
Де 0 - ви нічого не обираєте, та йдете на слідуюче завдання  
1 - обираєте функцію strpbrk, 2 - обираєте функцію strtol  
Введіть цифру: 1  
Введіть речення, але максимальна довжина його 100 символів: eirkepr doirh pirh  
Введіть символи для пошуку, але максимальна кількість 10: jb  
Робота прописаної функції  
Жодних відповідностей не знайдено.  
Робота вбудованої функції  
Жодних відповідностей не знайдено.

Введіть число для вибору функції  
Де 0 - ви нічого не обираєте, та йдете на слідуюче завдання  
1 - обираєте функцію strpbrk, 2 - обираєте функцію strtol  
Введіть цифру: 0

### Завдання №3

Введіть ім'я файлу (але максимальна кількість символів буде 3): quew  
Ви ввели ім'я більше за 3 символи  
Введіть ім'я файлу: que  
Введіть кількість слів у реченні: 3  
Файл input.txt було успішно відкрито  
Файл que\_out.dat було успішно відкрито

Ваш файл пустий тому програма не може обробляти ваші дані з файлу  
Будь ласка введіть дані у файл  
Для продовження натисніть будь-яку клавішу . . .

### Завдання №3

Введіть ім'я файлу (але максимальна кількість символів буде 3): que  
Введіть кількість слів у реченні: 3  
Файл input.txt було успішно відкрито  
Файл que\_out.dat було успішно відкрито

Дані були успішно записанні у вихідний файл  
1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 2

Дані з вихідного файлу:

Наші речення:  
lorem hello advbd n dtn  
rbwhrhb db dn  
erh dfgn  
wddfvsf

Результат дії:  
rbwhrhb db dn  
Всього 1 речень з 3 словами

1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 1

Дані з вхідного файлу:  
lorem hello advbd n dtn  
rbwhrhb db dn  
erh dfgn  
wddfvsf

1 - Програма виведе на екран дані з вхідного файлу

### Завдання №3

Введіть ім'я файлу (але максимальна кількість символів буде 3): que  
Введіть кількість слів у реченні: 4  
Файл input.txt було успішно відкрито  
Файл que\_out.dat було успішно відкрито

Дані були успішно записанні у вихідний файл  
1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 2

Дані з вихідного файлу:

Наші речення:  
lorem hello advbd n dtn  
rbwhrrhb db dn  
erh dfgn  
wddfvsf

Результат дії:  
Всього 0 речень з 4 словами

1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 1

Дані з вхідного файлу:  
lorem hello advbd n dtn  
rbwhrrhb db dn  
erh dfgn  
wddfvsf

1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: █



### Завдання №3

Введіть ім'я файлу (але максимальна кількість символів буде 3): que  
Введіть кількість слів у реченні: 5  
Файл input.txt було успішно відкрито  
Файл que\_out.dat було успішно відкрито

Дані були успішно записанні у вихідний файл  
1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 2

Дані з вихідного файлу:

Наші речення:  
lorem hello advbd n dtn  
rbwhrrhb db dn  
erh dfgn  
wddfvsf

Результат дії:  
lorem hello advbd n dtn  
Всього 1 речень з 5 словами

1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число: 1

Дані з вхідного файлу:  
lorem hello advbd n dtn  
rbwhrrhb db dn  
erh dfgn  
wddfvsf

1 - Програма виведе на екран дані з вхідного файлу  
2 - Програма виведе на екран дані з вихідного файлу  
будь-яке інше число, вихід з програми  
Введіть число:

## **6.Аналіз помилок**

Поки помилок не було

## **7.Висновок**

На мою думку ця лабораторна робота була дуже корисна та цікава, бо я навчився працювати із рядковими функціями, ознайомилися із бібліотекою cstring, ми ще раз повторили як працювати із файлами, як туди не перезаписувати інформацію, а ще і дописувати цю інформацію.

Навчився працювати із файлами проєкту, а саме як правильно реалізовувати заголовочний файл та як правильно реалізовувати вихідний файл проєкту.