Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації

і управління

Звіт

з лабораторної роботи № 9 з дисципліни

«Основи програмування-1

Базові конструкції»

«Рядки»

Варіант     5

Виконав студент       ІП-02 Геращенко Дмитро Ігорович

(шифр, прізвище, ім'я, по батькові)

Перевірив              Вітковська Ірина Іванівна

( прізвище, ім'я, по батькові)

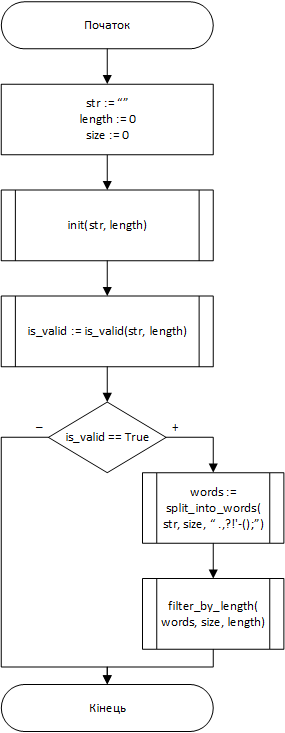
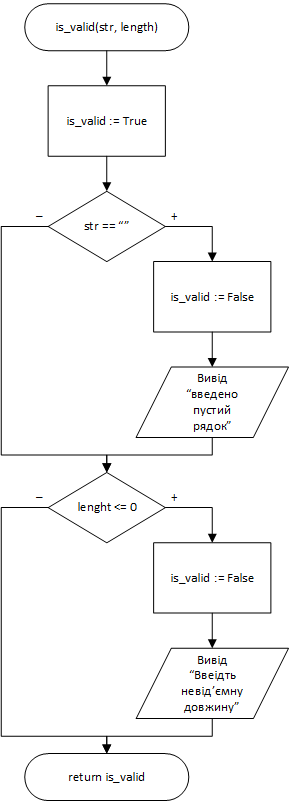
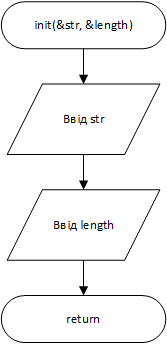
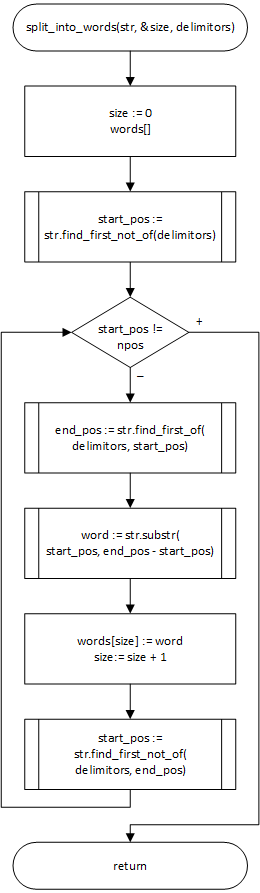
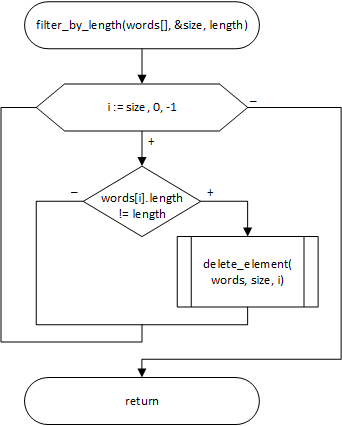
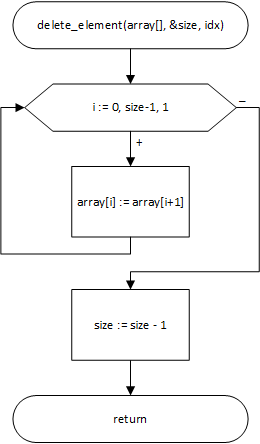
Київ 2020

# Лабораторна робота 9 Рядки Варіант 5

**Мета** – ознайомитися з особливостями реалізації текстових рядків, опанувати технологію їх використання, навчитися розробляти алгоритми та програми із застосуванням рядків.

1. Умова задачі

У заданому рядку символів визначити слова, довжина яких співпадає із заданою користувачем.

1. Блок-схема
2. Текст програми

Python:

##################################### main.py #####################################

from words import split\_into\_words, get\_by\_length, output

def init() -> tuple[str, int]:

    """ Приймає вхідні дані користувача та перевіряє їх на валідність. """

    s = input("Введіть ваш текст: ")

    if len(s) == 0: raise ValueError("Пустий рядок неприпустимий :(")

    length = int(input("Введіть необхідну довжину слів: "))

    if length <= 0: raise ValueError("Довжинна повинна бути більша нуля :(")

    return s, length

if \_\_name\_\_ == "\_\_main\_\_":

    s, length = init()

    words = split\_into\_words(s)

    output("Всі слова:", words)

    words = get\_by\_length(words, length)

    output("Відфільтровані слова:", words)

#################################### words.py #####################################

from string import punctuation

def split\_into\_words(s: str) -> list[str]:

    """ Розділяє рядок на слова враховуючи розділові знаки. """

    table = s.maketrans(punctuation, " "\*len(punctuation))

    return s.translate(table).split()

def get\_by\_length(words: list[str], length: int) -> list[str]:

    """ Повертає масив лише з тих слів, довжина яких задана. """

    return [word for word in words if len(word) == length]

def output(title: str, words: list[str]) -> None:

    """ Виводить масив слів через роздільник ", " """

    print(title)

    print(\*words, sep=", ")

C++:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main.cpp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream> // cin, cout, getline

#include <cstddef> // size\_t

#include <string> // string

#include "strings.h" // split\_into\_words\_s, filter\_by\_length, output

using namespace std;

void init(string\*, size\_t\*);

bool is\_valid(string, size\_t);

int main() {

    string str;

    size\_t length;

    init(&str, &length);

    if (is\_valid(str, length)) {

        size\_t size = 0;

        string \*words = split\_into\_words\_s(str, &size, " .,?!'-();");

        cout << "Всі слова:\n"; output(words, size);

        filter\_by\_length(words, &size, length);

        cout << "Відфільтровані слова:\n"; output(words, size);

        delete[] words;

    }

    return 0;

}

void init(string \*str, size\_t \*length) {

    cout << "Введіть ваш текст: "; getline(cin, \*str);

    cout << "Введіть необхідну довжину слів: "; cin >> \*length;

}

bool is\_valid(string str, size\_t length) {

    bool is\_valid = true;

    if (str.length() == 0) {

        cerr << "Пустий рядок неприпустимий :(" << endl;

        is\_valid = false;

    }

    if (length <= 0) {

        cerr << "Довжинна повинна бути більша нуля :(" << endl;

        is\_valid = false;

    }

    return is\_valid;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* strings.h \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstddef> // size\_t

#include <string> // string

using std::string, std::size\_t;

/\*\*

 \*  Розділяє рядок на слова або за заданими роздільниками (пробіл за замовчуванням).

 \*  @param str[in] Рядок, що буде розділений на слова.

 \*  @param size[out] К-сть розділених слів.

 \*  @param delimitors[in] Рядок з роздільниками.

 \*  @return Масив слів розділених пробілами в рядку.

 \*/

string \*split\_into\_words(string, size\_t\*);

string \*split\_into\_words(string, size\_t \*, string); // С-стиль

string \*split\_into\_words\_s(string, size\_t \*, string); // С++-стиль

/\*\*

 \*  Залишає в мисиві лише слова заданої довжини.

 \*  @param words[in, out] Масив зі словами, що буде відфільтрований.

 \*  @param size[in, out] К-сть слів у масиві.

 \*  @param lenght[int] Довжина слів, які потрібно залишити.

 \*/

void filter\_by\_length(string[], size\_t\*, size\_t);

// Виводить масив слів через роздільник ", "

void output(string[], size\_t);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* strings.cpp \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream> // cout

#include <sstream> // istringstream

#include <cstring> // size\_t, strtok, strcpy

#include <string> // string, npos

#include "arrays.h" // double\_array, delete\_element

using namespace std;

// Розділяє рядок на слова за пробілами

string \*split\_into\_words(string str, size\_t \*size) {

    istringstream ss(str);

    string word;

    \*size = 0;

    size\_t max\_size = (str.length() / 10) + 1;

    string \*words = new string[max\_size];

    while (ss >> word) {

        if (\*size > max\_size-1)

            double\_array(&words, &max\_size);

        words[(\*size)++] = word;

    }

    return words;

}

// Розділяє рядок на слова за заданими роздільниками (С-стиль)

string \*split\_into\_words(string str, size\_t \*size, string delimitors) {

    const char \*delims = delimitors.c\_str(); // Створуємо const рядок в С-стилі зі string

    char s[str.length()]; strcpy(s, str.c\_str()); // Створуємо mutable рядок в С-стилі зі string

    \*size = 0;

    size\_t max\_size = (str.length() / 10) + 1;

    string \*words = new string[max\_size];

    char \*ptr = strtok(s, delims);

    while (ptr != NULL) {

        if (\*size > max\_size-1)

            double\_array(&words, &max\_size);

        words[(\*size)++] = string(ptr);

        ptr = strtok(NULL, delims);

    }

    return words;

}

// Розділяє рядок на слова за заданими роздільниками (С++-стиль)

string \*split\_into\_words\_s(string str, size\_t \*size, string delimitors) {

    \*size = 0;

    size\_t max\_size = (str.length() / 10) + 1;

    string \*words = new string[max\_size];

    size\_t start\_pos = str.find\_first\_not\_of(delimitors),

           end\_pos;

    while (start\_pos != string::npos) {

        if (\*size > max\_size-1)

            double\_array(&words, &max\_size);

        end\_pos = str.find\_first\_of(delimitors, start\_pos);

        words[(\*size)++] = str.substr(start\_pos, end\_pos-start\_pos);

        start\_pos = str.find\_first\_not\_of(delimitors, end\_pos);

    }

    return words;

}

// Залишає в мисиві лише слова заданої довжини

void filter\_by\_length(string words[], size\_t \*size, size\_t length) {

    for (size\_t i = \*size; i-- > 0;) // Баг в привичному записі (переповення i)

        if (words[i].length() != length)

            delete\_element(words, size, i);

}

// Виводить масив слів через роздільник ", "

void output(string array[], size\_t size) {

    if (size > 0) {

        for (size\_t i = 0; i < size-1; i++)

            cout << array[i] << ", ";

        cout << array[size-1] << endl;

    }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* arrays.h \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <cstddef> // size\_t

using std::size\_t;

template <typename T>

void copy\_array(T array1[], T array2[], size\_t min\_size) {

    for (size\_t i = 0; i < min\_size; i++)

        array1[i] = array2[i];

}

template <typename T>

void double\_array(T \*\*array, size\_t \*max\_size) {

    T \*temp = new T[2 \* \*max\_size];

    copy\_array(temp, \*array, \*max\_size); // temp = array

    delete[] \*array;

    \*max\_size \*= 2;

    \*array = temp;

}

template <typename T>

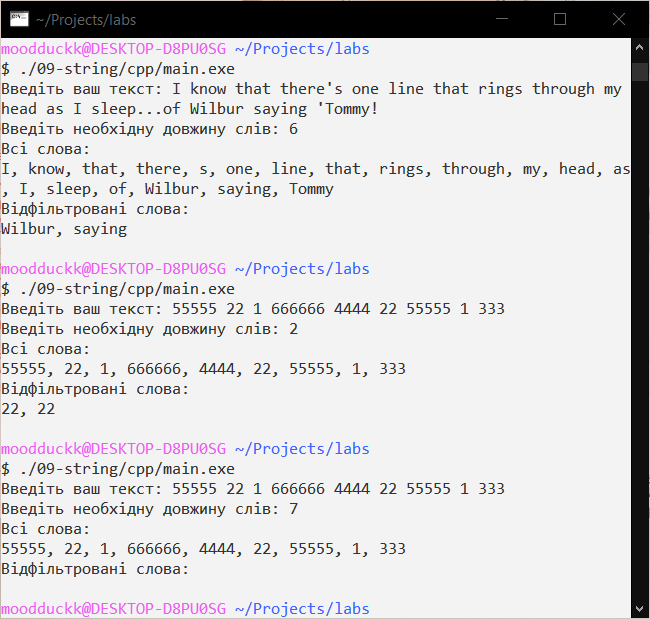
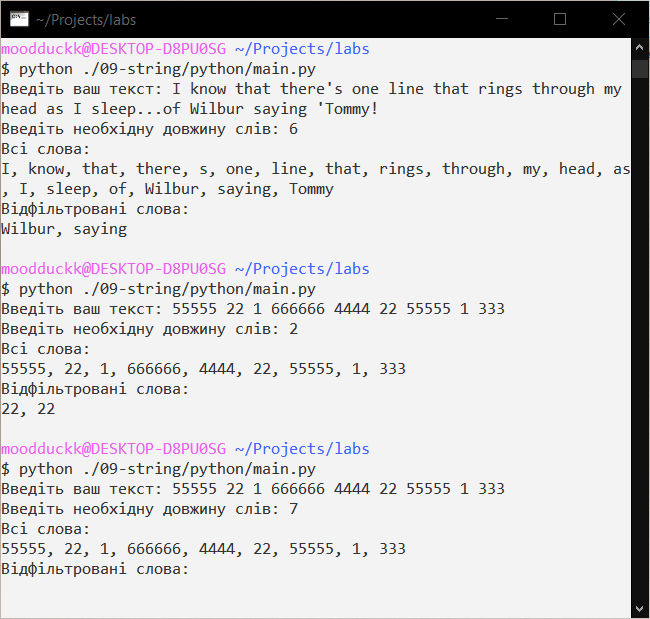
void delete\_element(T array[], size\_t \*size, size\_t idx) {

    for (size\_t i = idx; i < \*size-1; i++)

        array[i] = array[i+1];

    (\*size)--;

}

1. Копії екранних форм результатів роботи
2. Висновок

На цій лабораторній роботі було використано рядки та застосовано алгоритми їх обробки, такі як виділення лексем з рядка за розділювачами та знаходження довжини рядка. На мовах C++ та Python були дослідженні та застосовані базові алгоритми обробки рядків, та на їх основі написані програми за модульним підходом розробки.