# 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 № 9 з дисципліни

«Основи програмування-1.

Базові конструкції»

«Рядки»

Варіант 19

Виконав студент: ІП-11 Лисенко Андрій Юрійович

Перевірила: Вітковська Ірина Іванівна

Київ 2021

**Лабораторна робота №9**

**Рядки**

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

**Індивідуальне завдання**

**Варіант 19**

У заданому рядку символів знайти символи з максимальною і мінімальною частотою входження.

**Постановка задачі**

Для вирішення цієї задачі ініціалізуємо матрицю *ch\_mtx,* елементи якого будуть відповідати частоті входження певного символу. Для знаходження цієї матриці використаємо цикл в якому використаємо метод *find()* з класу *string* для знаходження певного символа. Також методом *erase()* видалимо цей елемент зі збереженого вхідного рядку, щоб запобігти багаторазового перелічування цього символу. Функції *get\_min\_arr()* та *get\_max\_arr()* знайдуть мінімальні та максимальні елементи матриці.

**Побудова математичної моделі**

Складемо таблицю змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Мінімум, Максимум | *int* | *min, max* | Зберігати мінімальну частоту входження символів |
| Розмір матриці символів | *int* | *ch\_len* | Зберігати розмір матриці символів |
| Мінімальний, максимальний розмір | *int* | *min\_len, max\_len* | Зберігати розмір масиву мінімальних, максимальних елементів |
| Рядок | *string* | *str* | Зберігати рядок |
| Довжина рядка | *int* | *str\_len* | Зберігати довжину рядка |
| Матриця символів | *int\*\** | *ch\_mtx* | Зберігати дані про частоту входження символів та їх положення |
| Масив мінімальних та максиальних символів | *char\** | *min\_arr, max\_arr* | Зберігати дані про елементи з мінімальною та максимальною частотою входження |

Складемо таблицю підпрограм:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Підпрограма | Тип | Ім’я | Параметри | Призначення |
| Знайти та видалити символи | *int* | *find\_del\_ch* | *str, ch* | Знайти та видалити всі входження вказаного символу в рядку |
| Декларація матриці | *int\*\** | *declr\_mtx* | *len* | Виділити пам'ять для матриці |
| Ініціалізація матриці символів | *void* | *get\_ch\_mtx* | *ch\_mtx, str, &len* | Знайти частоту входження кожного символу та записати у матрицю |
| Ініціалізація мінімального масиву | *void* | *get\_min\_arr* | *ch\_mtx, min\_arr, ch\_len, &min\_len, &min* | Знайти масив елементів з мінімальною частотою входження |
| Ініціалізація максимального масиву | *void* | *get\_max\_arr* | *ch\_mtx, max\_arr, ch\_len, &max\_len, &max* | Знайти масив елементів з максимальною частотою входження |
| Виведення матриці символів | *void* | *out\_ch\_mtx* | *ch\_mtx, len* | Вивести матрицю символів |
| Виведення матриці мінімальних та максимальних символів | *void* | *out\_min\_max* | *min\_arr, max\_arr, min, max, min\_len, max\_len* | Вивести масив мінімальних та максимальних символів |
| Видалення матриці | *void* | *del\_mtx* | *mtx, len* | Видалити матрицю з пам'яті |

**Псевдокод**

**Основна програма:**

**Початок**

**Введення** *str;*

*int\*\* ch\_mtx = declr\_mtx(str\_len);*

*char\* min\_arr = new char [str\_len];*

*char\* max\_arr = new char [str\_len];*

*get\_ch\_mtx(ch\_mtx, str, ch\_len);*

*out\_ch\_mtx(ch\_mtx, ch\_len);*

*get\_min\_arr(ch\_mtx, min\_arr, ch\_len, min\_len, min);*

*get\_max\_arr(ch\_mtx, max\_arr, ch\_len, max\_len, max);*

*out\_min\_max(min\_arr, max\_arr, min, max, min\_len, max\_len);*

*del\_mtx(ch\_mtx, str\_len);*

*delete[] min\_arr;*

*delete[] max\_arr;*

**Кінець**

**Підпрограма *declr\_mtx:***

**Початок *declr\_mtx***

*int\*\* mtx = new int\* [len];*

**Повторити для** *i* **від** *0* **до** *len***:**

*\*(mtx + i) = new int [2];*

**Все повторити**

**Кінець *declr\_mtx***

**Підпрограма *get\_ch\_mtx:***

**Початок *get\_ch\_mtx***

**Поки** *!str.empty()***:**

*ch = str[0];*

*ch\_mtx[i][0] = int(ch);*

*ch\_mtx[i][1] = find\_del\_ch(str, ch);*

*i++;*

**Все поки**

*len = i;*

**Кінець *get\_ch\_mtx***

**Підпрограма *find\_del\_ch:***

**Початок *find\_del\_ch***

*i = 0, frq = 0;*

**Поки** *i < str.length():*

**Якщо** *str[i] == ch* **то:**

*frq++;*

*str.erase(i, 1);*

**Інакше**

*i++;*

**Все якщо**

**Повернути** *frq;*

**Кінець *find\_del\_ch***

**Підпрограма *out\_ch\_mtx:***

**Початок *out\_ch\_mtx***

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Виведення** *ch\_mtx[i][1]*

**Все повторити**

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Виведення** *char(ch\_mtx[i][0])*

**Все повторити**

**Кінець *out\_ch\_mtx***

**Підпрограма *get\_min\_arr:***

**Початок *get\_min\_arr***

*min = ch\_mtx[0][1];*

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Якщо** *ch\_mtx[i][1] < min && !isspace(ch\_mtx[i][0])* **то:**

*min = ch\_mtx[i][1];*

**Все якщо**

**Все повторити**

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Якщо** *ch\_mtx[i][1] == min && !isspace(ch\_mtx[i][0])* **то:**

*min\_arr[min\_len] = char(ch\_mtx[i][0]);*

*min\_len++;*

**Все якщо**

**Все повторити**

**Кінець *get\_min\_arr***

**Підпрограма *get\_max\_arr:***

**Початок *get\_max\_arr***

*max = ch\_mtx[0][1];*

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Якщо** *ch\_mtx[i][1] > max && !isspace(ch\_mtx[i][0])* **то:**

*max = ch\_mtx[i][1];*

**Все якщо**

**Все повторити**

**Повторити для** *i* **від** *0* **до** *ch\_len***:**

**Якщо** *ch\_mtx[i][1] == max && !isspace(ch\_mtx[i][0])* **то:**

*max\_arr[max\_len] = char(ch\_mtx[i][0]);*

*max\_len++;*

**Все якщо**

**Все повторити**

**Кінець *get\_max\_arr***

**Підпрограма *out\_min\_max;***

**Початок *out\_min\_max***

**Виведення** *" The character(s) with minimum amount of occurance (except of spaces) is";*

**Повторити для** *i* **від** *0* **до** *min\_len***:**

**Виведення** *min\_arr[i];*

**Все повторити**

**Виведення** *" The character(s) with maximum amount of occurance (except of spaces) is";*

**Повторити для** *i* **від** *0* **до** *max\_len***:**

**Виведення** *max\_arr[i];*

**Все повторити**

**Кінець *out\_min\_max***

**Підпрограма *del\_mtx:***

**Початок *del\_mtx***

**Повторити для** *i* **від** *0* **до** *min\_len***:**

*delete[] \*(mtx + i);*

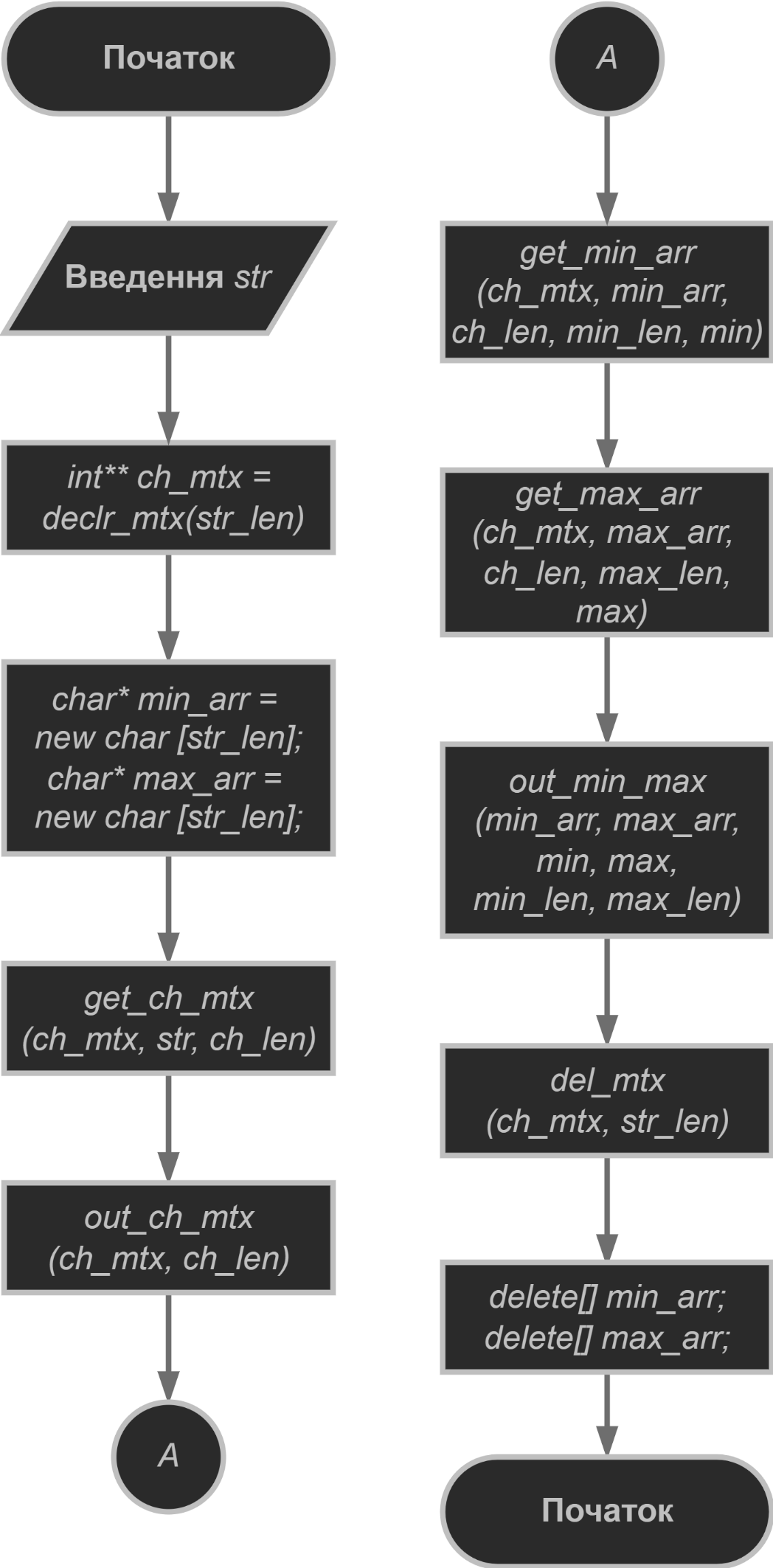
**Все повторити**

*delete[] mtx;*

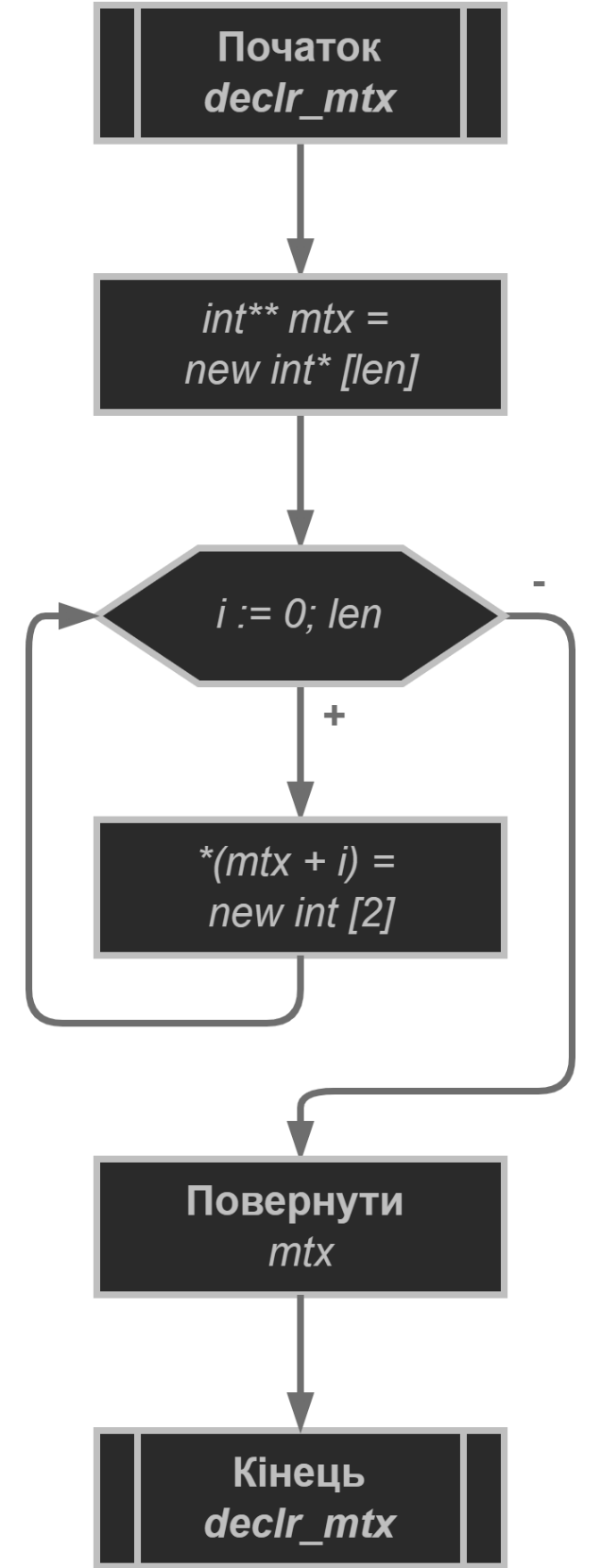
**Кінець *del\_mtx***

**Блок-схема**

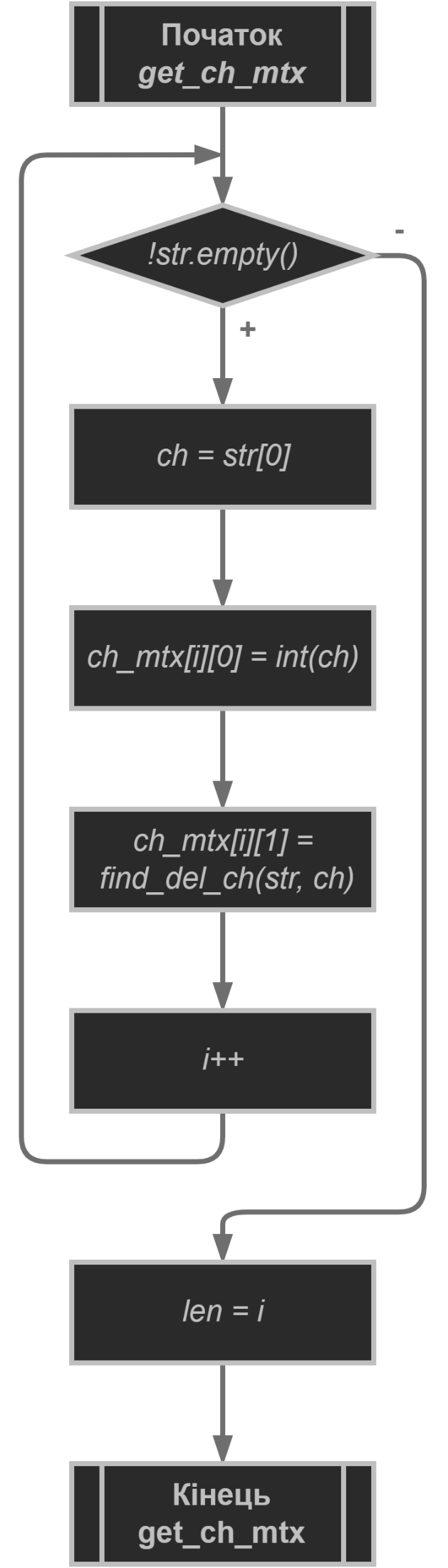
**Основна програма**



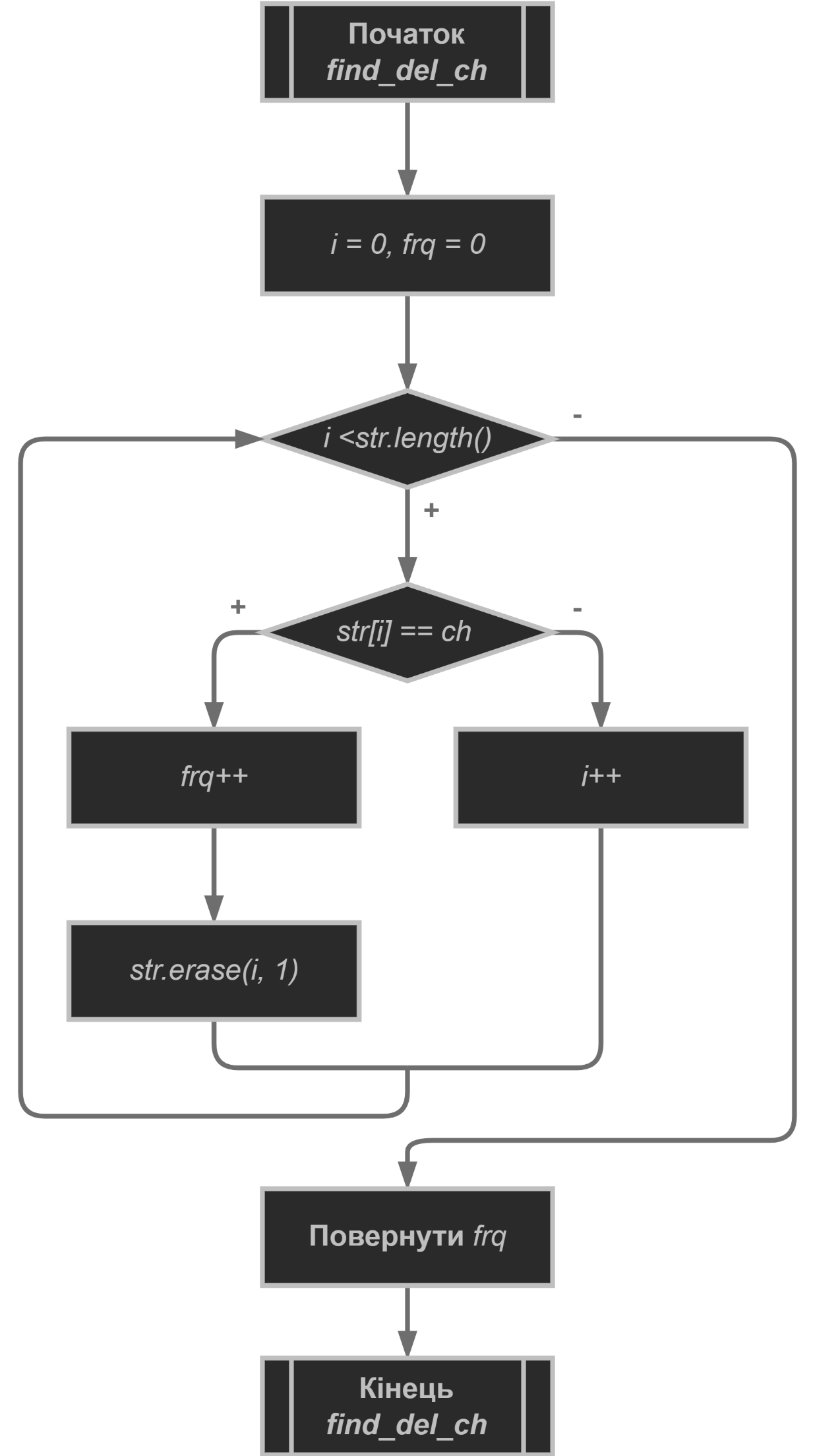
**Підпрограма *declr\_mtx***



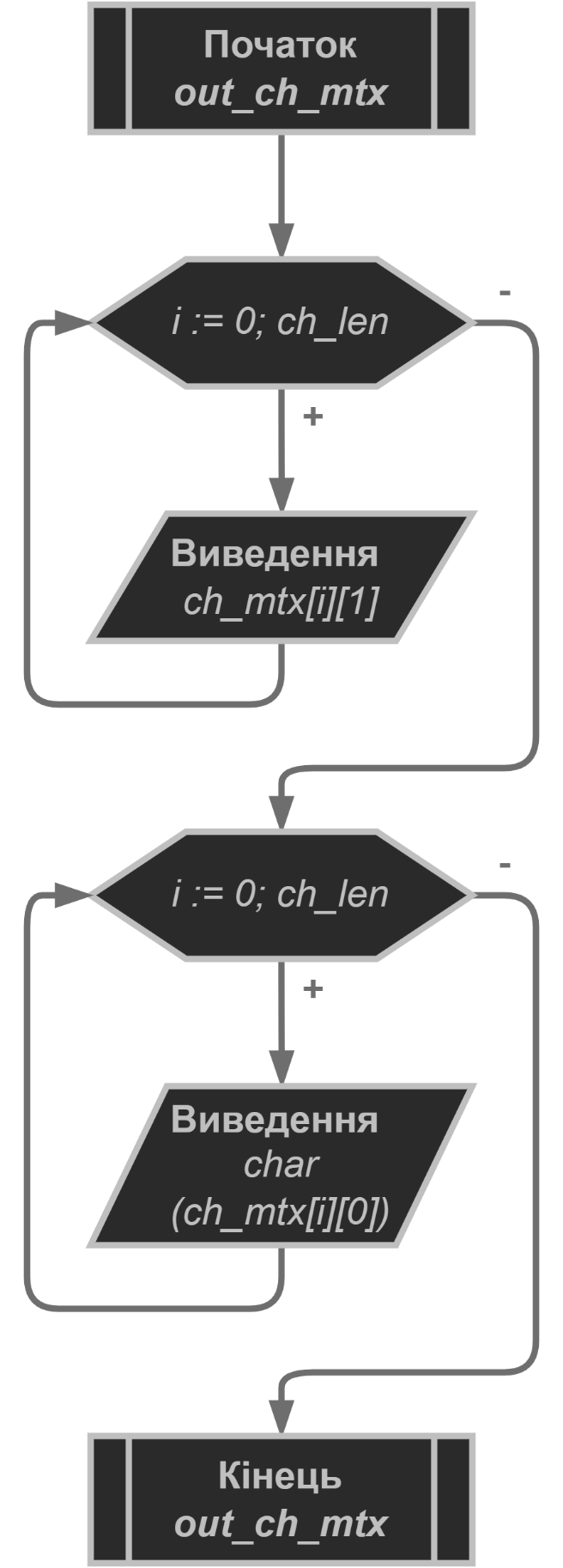
**Підпрограма *get\_ch\_mtx***



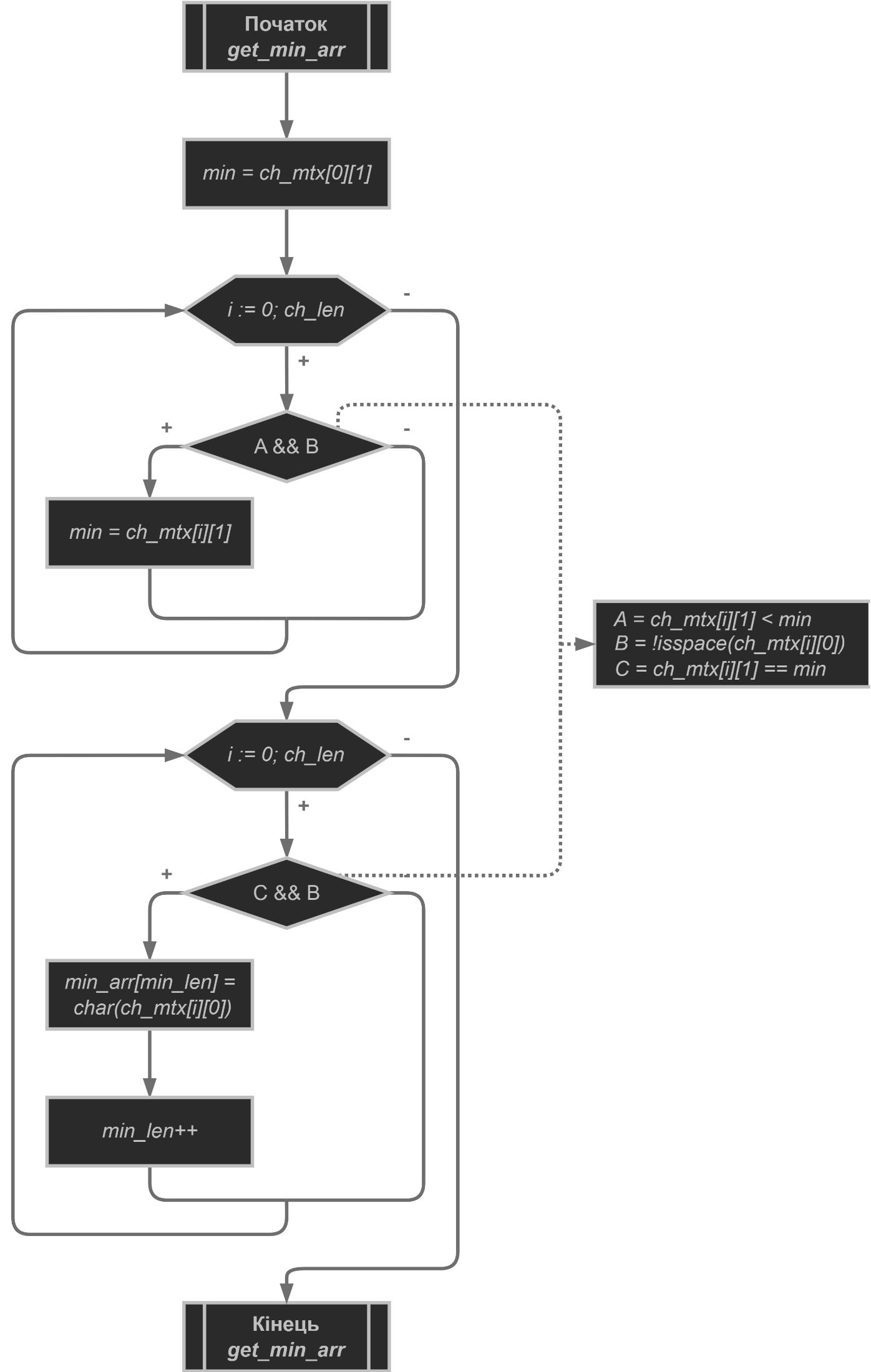
**Підпрограма *find\_del\_ch***



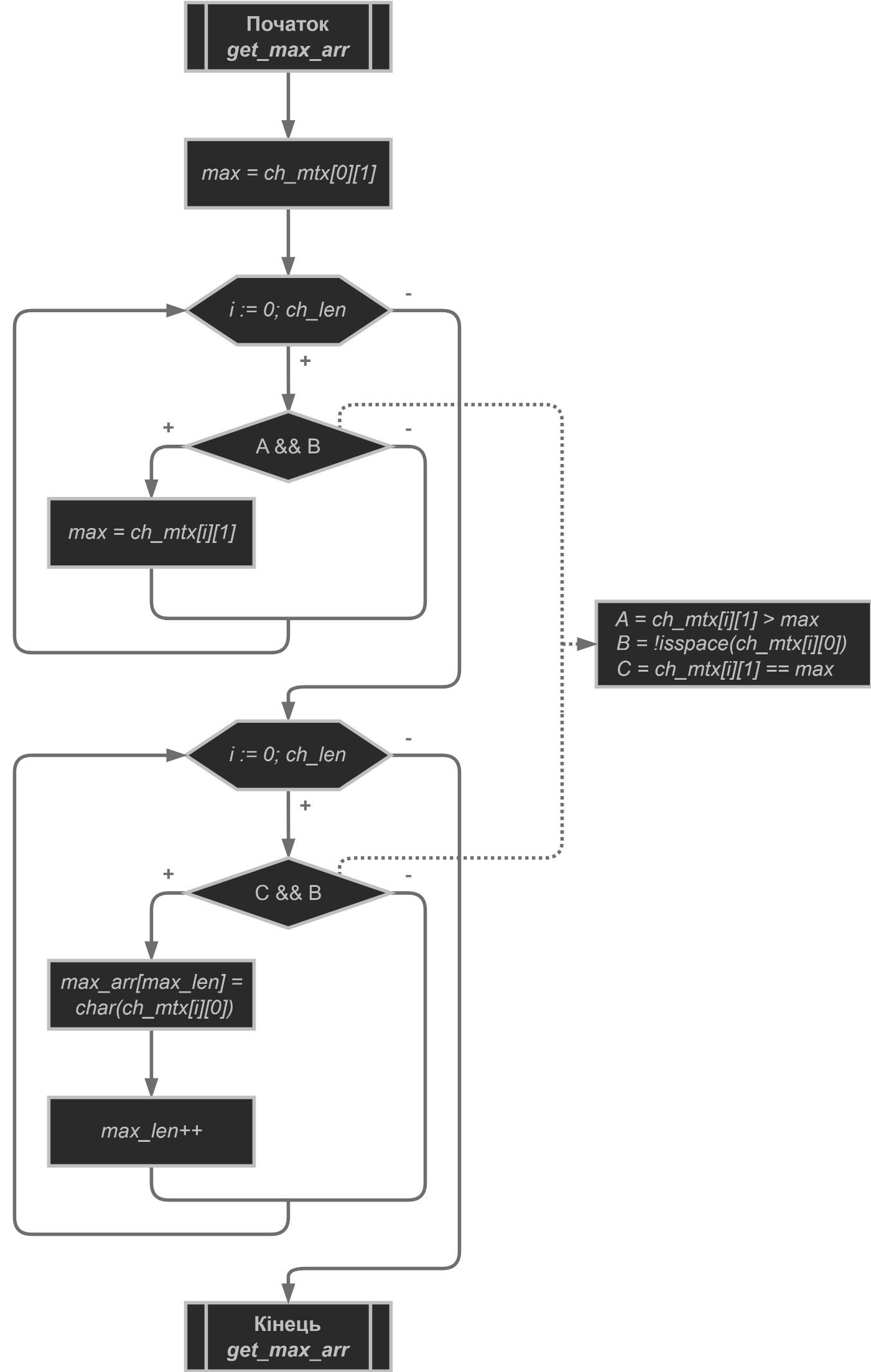
**Підпрограма *out\_ch\_mtx***



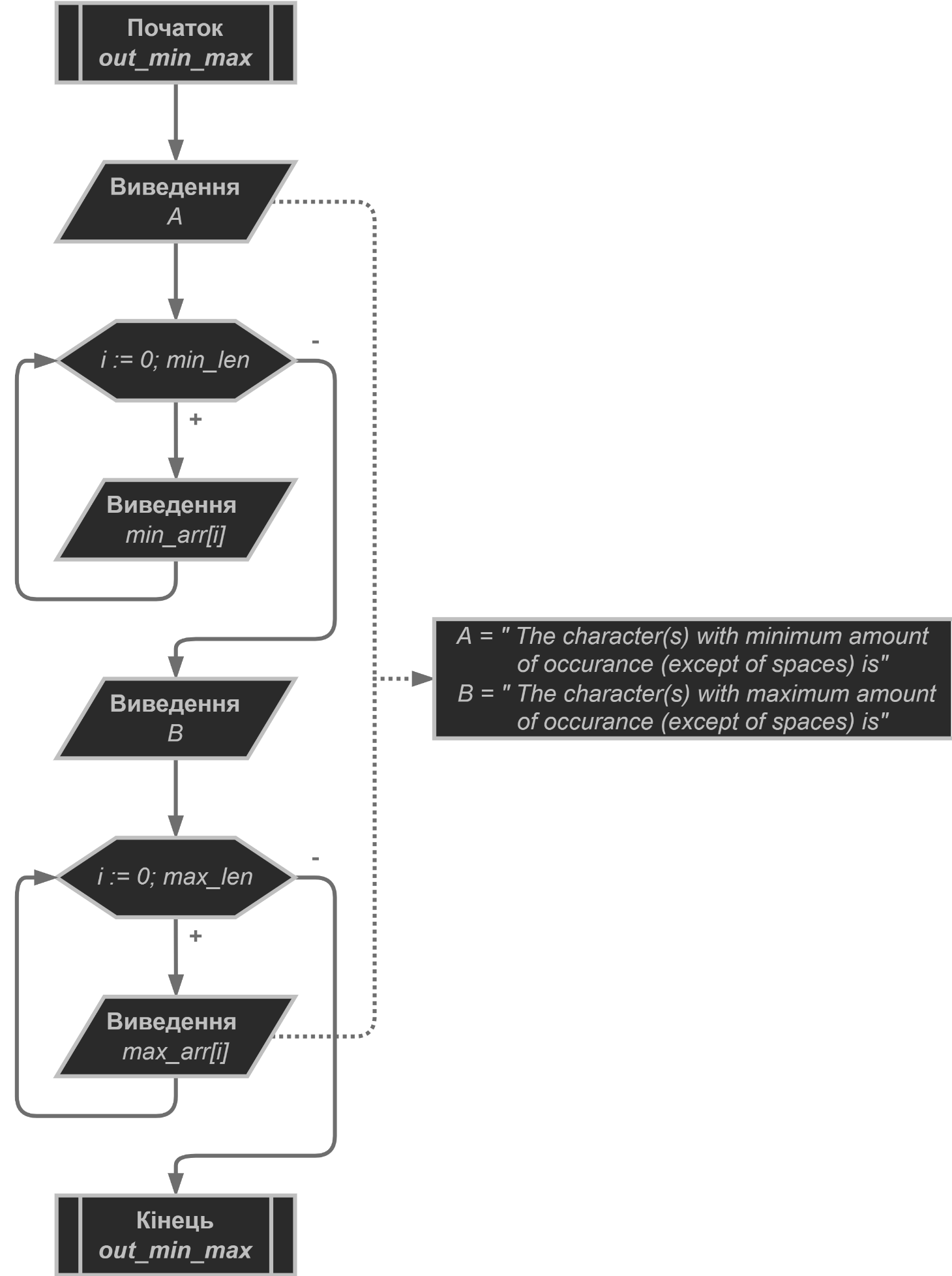
**Підпрограма *get\_min\_arr***

****

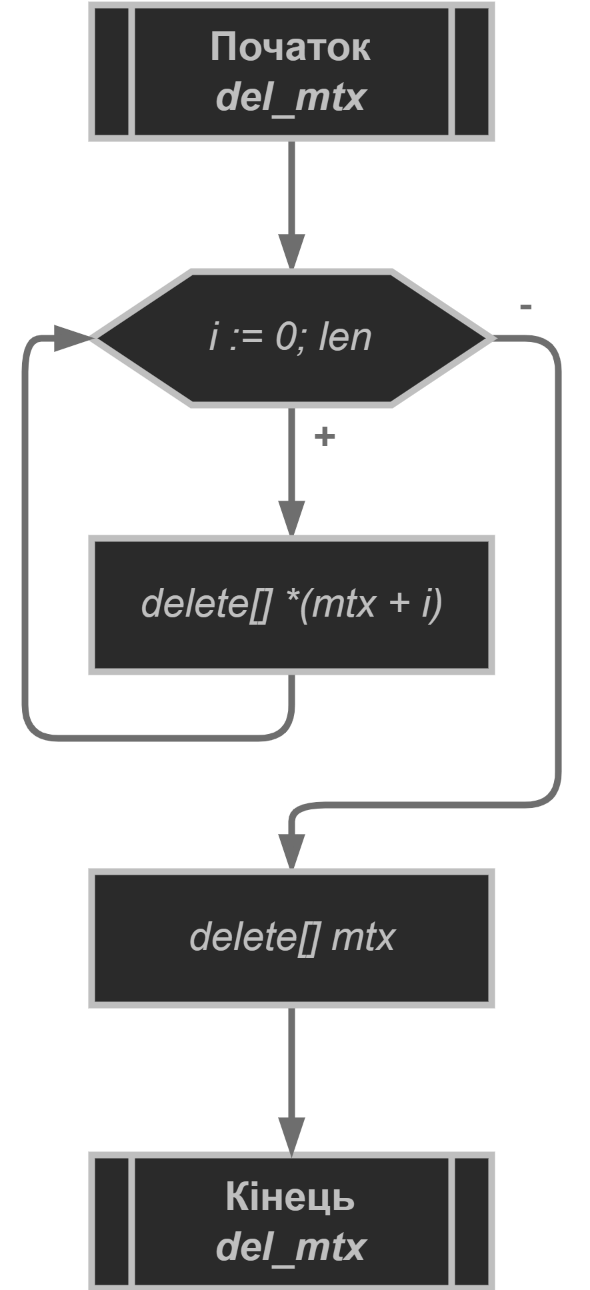
**Підпрограма *get\_max\_arr***



**Підпрограма *out\_min\_max***



**Підпрограма *del\_mtx***

****

**Код на С++**

#include <iostream>

#include <iomanip>

#include <string>

int find\_del\_ch(std::string&, char);

int\*\* declr\_mtx(int);

void get\_ch\_mtx(int\*\*, std::string, int&);

void get\_min\_arr(int\*\*, char\*, int, int&, int&);

void get\_max\_arr(int\*\*, char\*, int, int&, int&);

void out\_ch\_mtx(int\*\*, int);

void out\_min\_max(char\*, char\*, int, int, int, int);

void del\_mtx(int\*\*, int);

int main() {

std::string str;

std::cout << "Enter string: "; getline(std::cin, str);

int min, max, ch\_len, min\_len, max\_len, str\_len = str.size();

int\*\* ch\_mtx = declr\_mtx(str\_len);

char\* min\_arr = new char [str\_len];

char\* max\_arr = new char [str\_len];

get\_ch\_mtx(ch\_mtx, str, ch\_len);

out\_ch\_mtx(ch\_mtx, ch\_len);

get\_min\_arr(ch\_mtx, min\_arr, ch\_len, min\_len, min);

get\_max\_arr(ch\_mtx, max\_arr, ch\_len, max\_len, max);

out\_min\_max(min\_arr, max\_arr, min, max, min\_len, max\_len);

del\_mtx(ch\_mtx, str\_len);

delete[] min\_arr;

delete[] max\_arr;

system("pause");

return 0;

}

int\*\* declr\_mtx(int len) {

int\*\* mtx = new int\* [len];

for (int i = 0; i < len; i++)

\*(mtx + i) = new int [2];

return mtx;

}

void get\_ch\_mtx(int\*\* ch\_mtx, std::string str, int &len) {

char ch, i;

for (i = 0; !str.empty(); i++) {

ch = str[0];

ch\_mtx[i][0] = int(ch);

ch\_mtx[i][1] = find\_del\_ch(str, ch);

}

len = i;

}

int find\_del\_ch(std::string &str, char ch) {

int i = 0, frq = 0;

while (i < str.length())

if (str[i] == ch) {

frq++;

str.erase(i, 1);

} else {

i++;

}

return frq;

}

void out\_ch\_mtx(int\*\* ch\_mtx, int ch\_len) {

int width;

std::cout << "\nCharacter frequency list:\n[ ";

for (int i = 0; i < ch\_len; i++)

std::cout << " " << ch\_mtx[i][1] << " ";

std::cout << "]\n";

std::cout << "[ ";

for (int i = 0; i < ch\_len; i++) {

width = (int)log10(ch\_mtx[i][1]) + 2;

std::cout << std::setw(width) << "'" << char(ch\_mtx[i][0]) << "' ";

}

std::cout << "]\n";

}

void get\_min\_arr(int\*\* ch\_mtx, char\* min\_arr, int ch\_len, int &min\_len, int &min) {

min = ch\_mtx[0][1];

for (int i = 1; i < ch\_len; i++)

if (ch\_mtx[i][1] < min && !isspace(ch\_mtx[i][0]))

min = ch\_mtx[i][1];

min\_len = 0;

for (int i = 0; i < ch\_len; i++)

if (ch\_mtx[i][1] == min && !isspace(ch\_mtx[i][0])) {

min\_arr[min\_len] = char(ch\_mtx[i][0]);

min\_len++;

}

}

void get\_max\_arr(int\*\* ch\_mtx, char\* max\_arr, int ch\_len, int &max\_len, int &max) {

max = ch\_mtx[0][1];

for (int i = 1; i < ch\_len; i++)

if (ch\_mtx[i][1] > max && !isspace(ch\_mtx[i][0]))

max = ch\_mtx[i][1];

max\_len = 0;

for (int i = 0; i < ch\_len; i++)

if (ch\_mtx[i][1] == max && !isspace(ch\_mtx[i][0])) {

max\_arr[max\_len] = char(ch\_mtx[i][0]);

max\_len++;

}

}

void out\_min\_max(char\* min\_arr, char\* max\_arr, int min, int max, int min\_len, int max\_len) {

std::cout << "\nThe character(s) with minimum amount of occurance (except of spaces) is ";

for (int i = 0; i < min\_len; i++) {

std::cout << "'" << min\_arr[i] << "' ";

}

std::cout << ": " << min << " time(s)\n";

std::cout << "The character(s) with maximum amount of occurance (except of spaces) is ";

for (int i = 0; i < max\_len; i++) {

std::cout << "'" << max\_arr[i] << "' ";

}

std::cout << ": " << max << " time(s)\n\n";

}

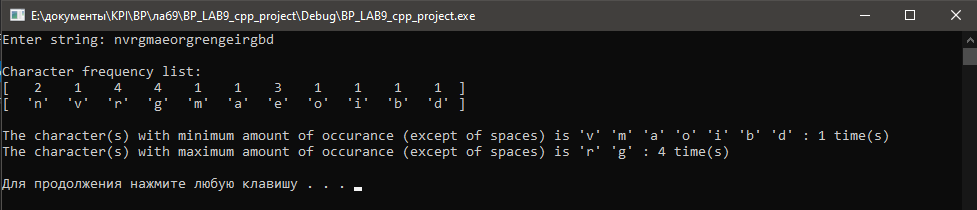
void del\_mtx(int\*\* mtx, int len) {

for (int i = 0; i < len; i++)

delete[] \*(mtx + i);

delete[] mtx;

}

****

**Висновок**

Ми використали функцію get\_ch\_mtx для знаходження матриці частоти входження символів, у якій лінійним пошуком знайшли частоту входження певного символу за допомогою підпрограми find\_del\_ch(). Також ми ініціювали масиви min\_arr, max\_arr які зберігали символи з мінімальною та максимальною частотою входження. Отже, ми ознайомилися з особливостями реалізації текстових рядків, опанували технологію їх використання та навчилися розробляти алгоритми та програми із застосуваннням рядків.