

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

КАФЕДРА «Обчислювальної техніки та програмування»

Лабораторна робота №8-10

Тема: «Вступ до документації проекту»

Виконала :

ст. гр. КІТ-120В, Семенченко Вікторія;

Перевірив:

Бульба С.С.

Харків 2020

**Мета:** для кожної розробленої функції, що були виконані у попередній роботі, слід зробити схему алгоритмів, додати можливість генерації Doxygen документації, розробити повноцінний звіт для лабораторної роботи «Функції» у двох форматах.

1. Блок-схеми алгоритмів для кожної розробленої функції, що були виконані у попередній роботі

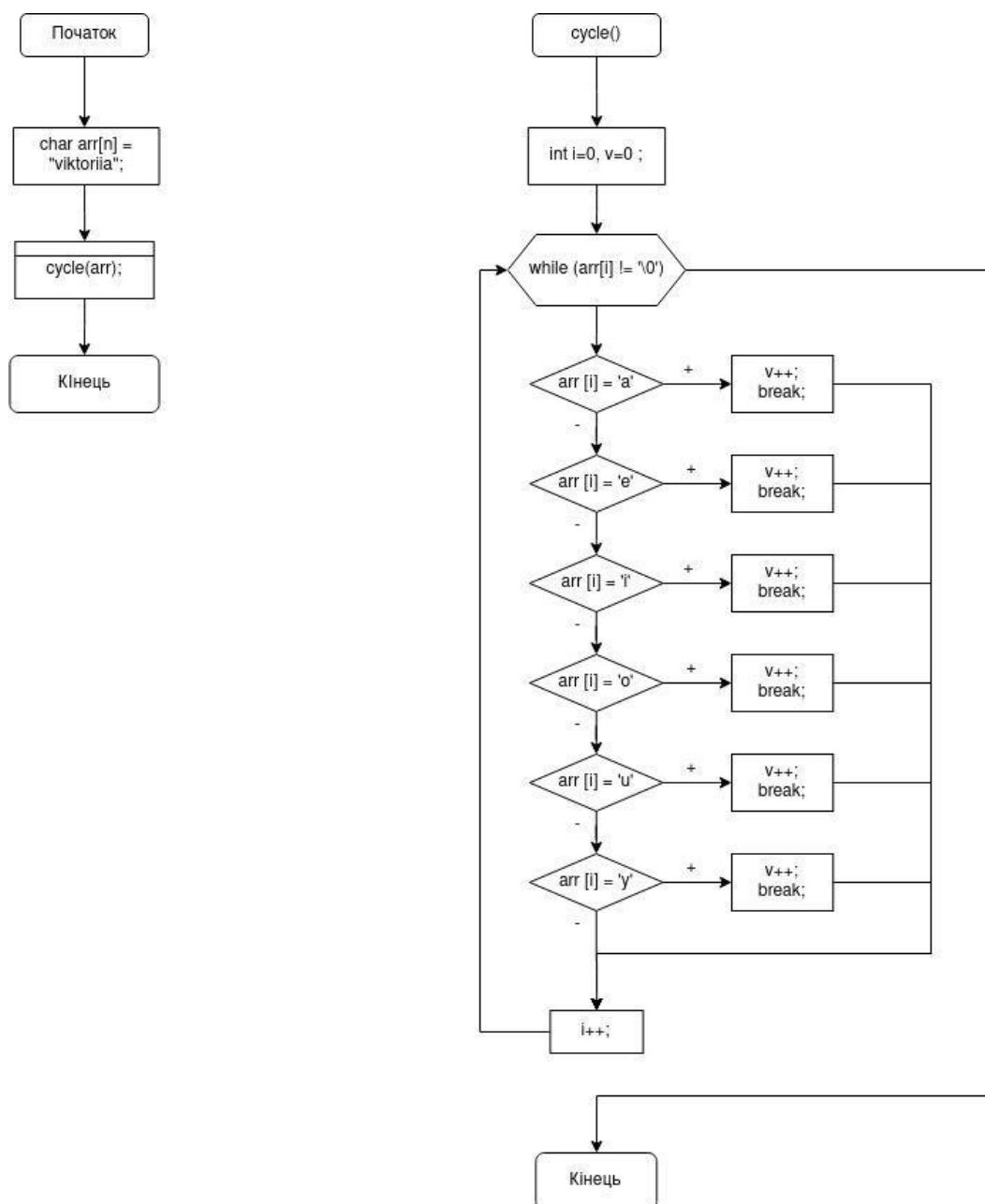


Рисунок 1.1 – блок-схема до завдання 1

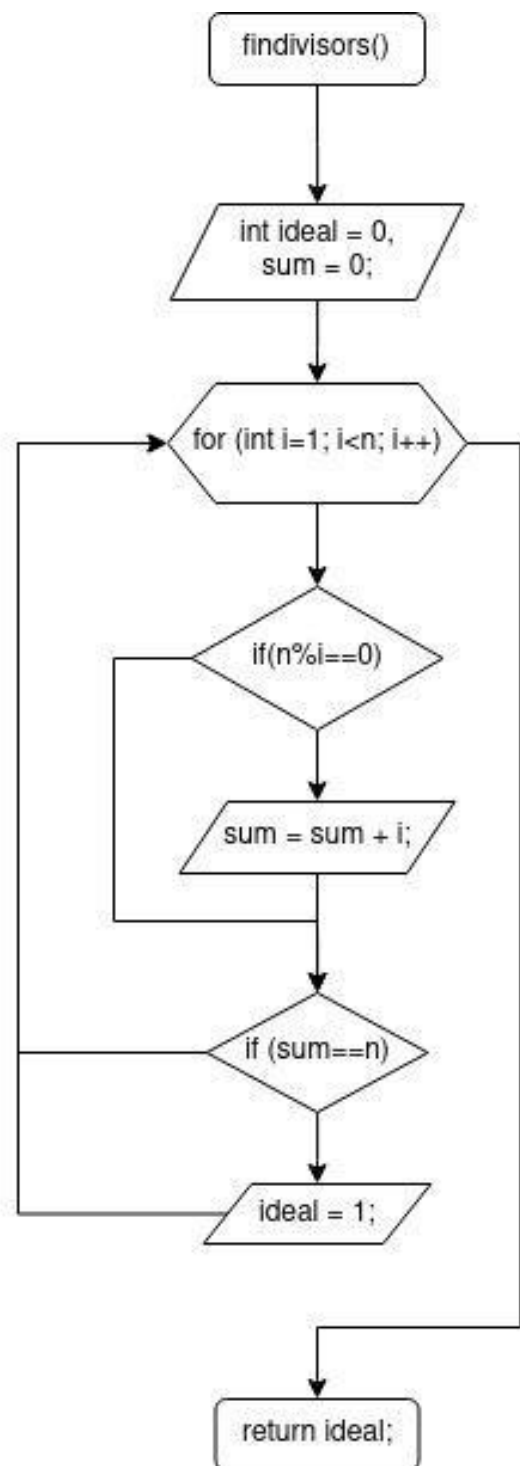
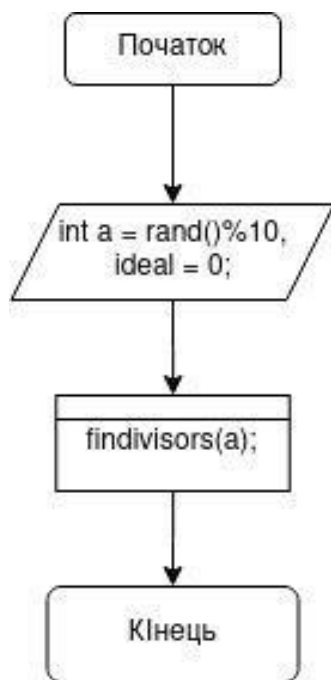


Рисунок 1.2 – блок-схема до завдання 2

## 2. Створення можливості генерації Doxygen документації

```
1 /**
2  * @mainpage
3  * # Загальне завдання
4  * 1. **Визначити**, чи є задане число простим
5  * @author Semenchenko V.
6  * @date 20-dec-2020
7  * @version 1.0
8  */
9
10 /**
11  * @file main.c
12  * @brief Файл з демонстрацією роботи структури та методів
13  * оперування ним.
14  *
15  * @author Semenchenko V.
16  * @date 20-dec-2020
17  * @version 1.0
18  */
19
20 #include <stdio.h>
21
22 /**
23  * @function finddivisors
24  *
25  * Функція для визначення чи є задане число простим
26  * Послідовність дій:
27  * - Цикл від 0 до n:
28  *   - Якщо  $n \% i = 0$ , то  $sum = sum + i$ ;
29  * - Якщо  $sum = n$ , то  $ideal = 1$ ;
30  * @return успішний код повернення результату роботи функції (ideal)
31  */
```

Рисунок 2.1 – приклад оформлення Doxygen документації

```
1 ## Common Project info
2 PROJECT_NAME = "Лабораторна робота 07"
3 PROJECT_BRIEF = "Визначити, чи є число простим"
4 PROJECT_NUMBER = 7.1(5)
5
6 ## output options
7 OUTPUT_DIRECTORY = ./dist
8 OUTPUT_LANGUAGE = Ukrainian
9 DOXYFILE_ENCODING = UTF-8
10
11 ## input options
12 INPUT_ENCODING = UTF-8
13 RECURSIVE = YES
14 FILE_PATTERNS = *.c *.h *.md
15 #####
16 #USE_MDFILE_AS_MAINPAGE = README.md
17 #####
18
19
20
21 ## processing options
22 EXTRACT_ALL = YES
23 EXTRACT_PRIVATE = YES
24 EXTRACT_STATIC = YES
25 JAVADOC_AUTOBRIEF = YES
26 OPTIMIZE_OUTPUT_FOR_C = YES
27
28 ## Draw calls graph + object relations (for cpp)
29 HAVE_DOT = YES
30 DOT_PATH = /usr/local/bin/dot
31 UML_LOOK = YES
32 TEMPLATE_RELATIONS = YES
33 CALL_GRAPH = YES
34
35 ## Customized generate options
36 GENERATE_HTML = YES
37 GENERATE_LATEX = YES
```

Рисунок 2.2 – заповнення Doxyfile для коректної генерації Doxygen

# Лабораторна робота 07

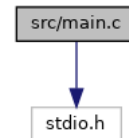
Визначити, чи є число простим

Титульна сторінка	Додаткова інформація	Файли ▾
src		
Файл main.c		

Файл з демонстрацією роботи структури та методів оперування ним. [Детальніше...](#)

```
#include <stdio.h>
```

Діаграма включених заголовочних файлів для main.c:



## Функції

int **finddivisors** (int n)  
@function finddivisors [Детальніше...](#)

int **main** ()  
Головна функція. [Детальніше...](#)

## Детальний опис

Рисунок 2.3 – сгенерована Doxygen документація

## 3. Написання звіту у форматі Markdown

```
1 # Загальне завдання
2 1. Реалізувати програму відповідно до індивідуального завдання.
3
4 ##Індивідуальне завдання
5 1. **Визначити** кількість голосних букв у слові
6
7 **ВИКОНАННЯ РОБОТИ**
8 2.1 Функціональне призначення
9     Програма призначена для визначення кількості голосних у слові
10 2.2 Створення репозиторію
11     Створивши репозиторій на github, клоную його та створюю всі файли за вимогами структури проекту.
12 2.3 Написання коду програми
13 ...
14 int main (){
15     char arr[n] = "viktoria";
16     cycle(arr);
17     return 0;
18 }
19
20 int cycle(char arr[n]){
21     int i=0, v=0 ;
22     while (arr[i] != '\0') {    // create a cycle for counting vowels
23         switch (arr [i]){
24             case 'a': v++; break;
25             case 'e': v++; break;
26             case 'i': v++; break;
27             case 'o': v++; break;
28             case 'u': v++; break;
29             case 'y': v++; break;
30         }
31         i++;
32     }
33 }
34 ...
35 ![lab06.png](./doc/lab06.png)
```

Рисунок 3.1 – звіт у форматі Markdown

## Загальне завдання

1. Реалізувати програму відповідно до індивідуального завдання.

##Індивідуальне завдання

1. Визначити кількість голосних букв у слові

**ВИКОНАННЯ РОБОТИ** 2.1 Функціональне призначення Програма призначена для визначення кількості голосних у слові 2.2 Створення репозиторію Створивши репозиторій на github, клоную його та створюю всі файли за вимогами структури проекту. 2.3 Написання коду програми

```
int main (){\n    char arr[n] = "viktoriiia";\n    cycle(arr);\n    return 0;\n}\n\nint cycle(char arr[n]){ \n    int i=0, v=0 ;\n    while (arr[i] != '\\0') {    // create a cycle for counting vowels\n        switch (arr [i]){ \n            case 'a': v++; break;\n            case 'e': v++; break;\n            case 'i': v++; break;\n            case 'o': v++; break;\n            case 'u': v++; break;\n            case 'y': v++; break;\n        }\n        i++;\n    }\n}
```

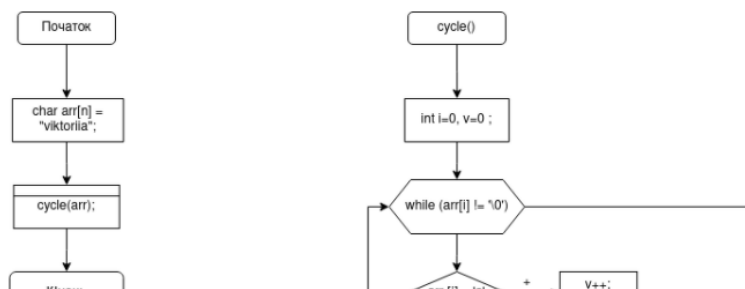
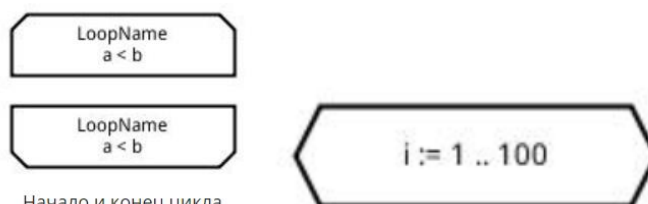


Рисунок 3.2 – звіт у форматі Markdown на гітхабі

## 4. Відповіді на питання :

1) Відповіді на запитання лабораторної роботи №8.

1. Схематичний опис необхідний для того, щоб наглядно показати алгоритм роботи коду.
2. Введення/виведення даних, дія, функція, цикл, умова.



3. Начало и конец цикла

4. Існують спеціальні програми для створення блок-схем в ручну та автоматично за кодом програми.

2) Відповіді на запитання лабораторної роботи №9.

1. Doxygen — кросплатформна система документування початкового коду програм.
2. Doxygen має вбудовану підтримку генерації документації в форматі HTML, LaTeX, man, RTF і XML.
3. Прописати команду doxygen.
4. Dot - консольна програма що розташовує граф на площині і виводить його зображення в різні графічні формати (PostScript, PDF, SVG, та інші).
5. Mscgen – невелика програма, яка аналізує опис Message Sequence.
6. /\*\* початок \*/ кінець
7. Doxygen – генерує докциджен документацію.

**Висновок:** Я зробила схему алгоритмів, додала можливість генерації Doxygen документації, розробила повноцінний звіт для лабораторної роботи «Функції» у двох форматах для кожної розробленої функції, що були виконані у попередній роботі.