

Лабораторна робота № 9. Вступ до документації коду

1. Вимоги

1.1 Розробник: Студент групи КІТ-120а Лисенко Микола Валерійович;

1.2 Перевірив: Асистент Челак Віктор Володимирович;

1.3 Загальне завдання: Розробити повноцінний звіт для лабораторної роботи "Функції".

2. Загальний опис проекту.

2.1 Проект складається з

Папки src з вихідними кодами до 3 завдань, Doxyfile, Makefile, Readme.

2.2 Структура проекту

```
├── Doxyfile
├── Makefile
├── README.md
└── src
    ├── lab05_01.c
    ├── lab06_04.c
    └── lab07_01.c
```

3. Опис коду lab05_01.

3.1 Функціональне призначення

Програма визначає найбільші спільні дільники пар чисел. Результат зберігається в змінних result_1, result_2, result_3. Демонстрація результату передбачає покрокове виконання програми.

3.2 Опис логічної структури

Спочатку задаються числа за допомогою генератора псевдовипадкових чисел. Після чого викликається функція gcd, котра повертає НСД двох чисел, виконання якої зберігається у змінну.

3.3 Головна функція main()

Схема алгоритму подана на рисунку 1.

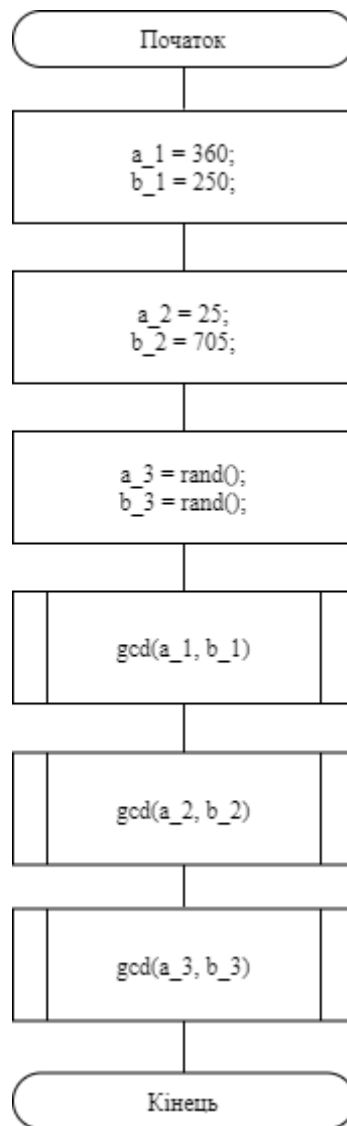


Рисунок 1 – Схема алгоритму головної функції main().

3.4 Функція gcd()

Визначення найбільшого спільного дільника двох чисел. Схема алгоритму зображена на рисунку 2.

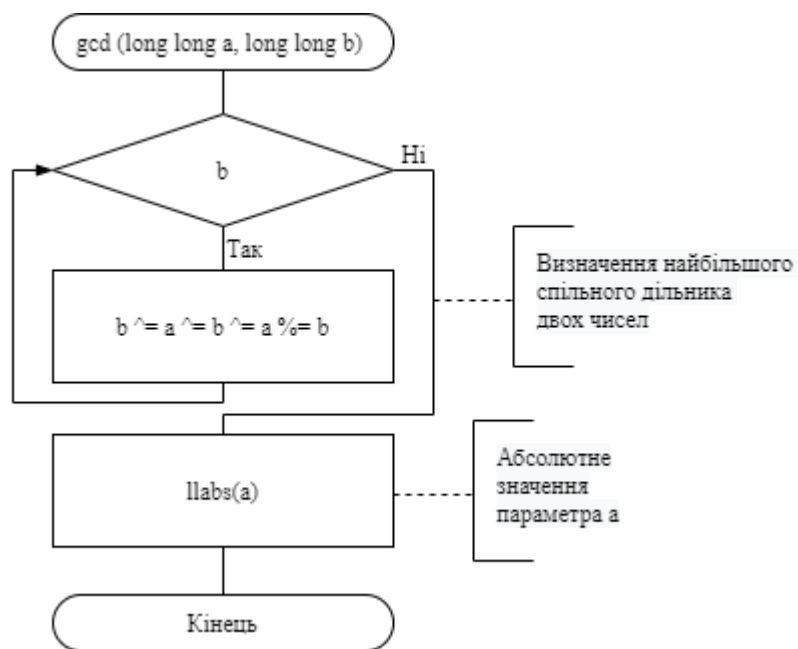


Рисунок 2 – Схема алгоритму функції gcd().

3.5 Результат виконання програми.

Використавши debug можна перевірити програму на дієздатність. Результат на рисунку 3.

a_1	360
b_1	250
a_2	25
b_2	705
a_3	750146112
b_3	1622906212
result_1	10
result_2	5
result_3	4

Рисунок 3 – результат виконання програми.

4. Опис коду lab06_04.c

4.1 Функціональне призначення

Визначення кількості слів, розділених пробілами, у заданому рядку. Результат зберігається у змінних result_1, result_2, result_3. Демонстрація результату передбачає покрокове виконання програми.

4.2 Опис логічної структури

Виконується оголошення речення випадкової довжини, яке заповнюється випадковими символами від 'а' до 'z' та пробілами. За допомогою функції `wnum` визначаємо кількість слів, розділених пробілами у заданому рядку.

4.3 Головна функція `main()`

Схема алгоритму подана на рисунку 4.

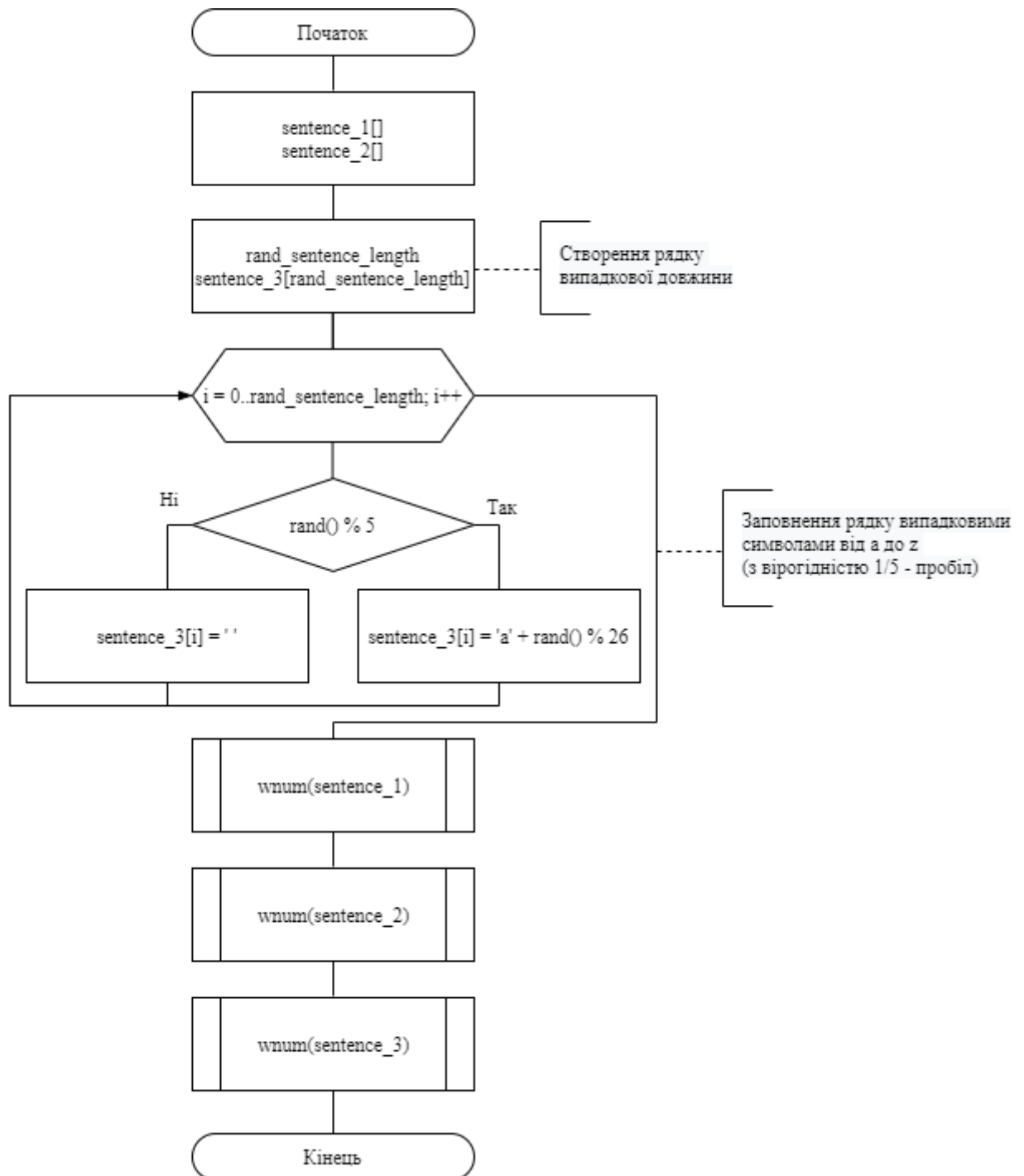


Рисунок 4 – Схема алгоритму головної функції `main()`.

4.4 Функція `wnum`

Визначає кількість слів, розділених пробілами у заданому рядку. Схема алгоритму на рисунку 5.

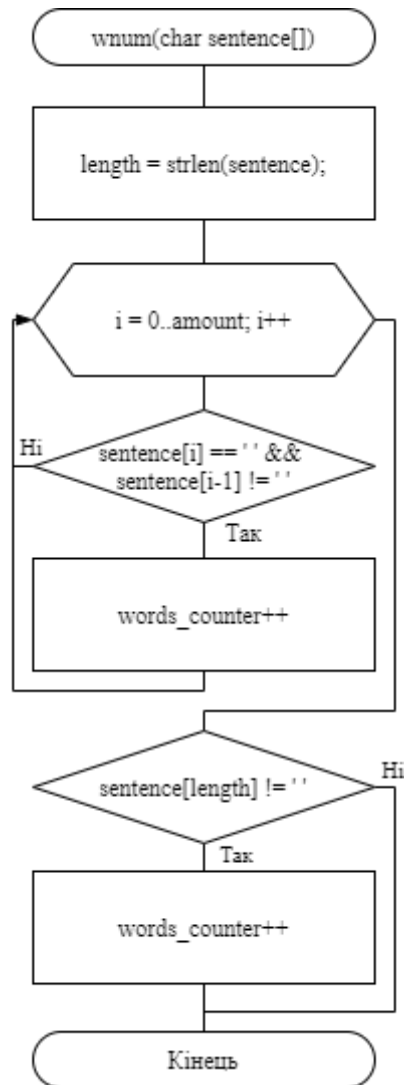


Рисунок 5 – Схема алгоритму функції `wnum()`.

4.5 Результат виконання програми.

Використавши `debug` можна перевірити програму на дієздатність. Результат на рисунку 3.

sentence_1	"dsffdssdf sdfsd fsd fsd"
sentence_2	"fssdfs dsfghg hdsff reg gfhg wae gf fsd trh trgdf ery hf fdf sdf"
rand_sentence_length	75
sentence_3	0x7ffffffeac0 "m l z krqbacnshcuye hvxaco dmau xiuiuh iwxbm mqgwy nqxot qsbg dpdw bhxty"
result_1	5
result_2	14
result_3	13

Рисунок 6 – результат виконання програми.

5. Опис коду lab07_01.c

5.1 Функціональне призначення

Програма визначає кількість пар у послідовності, де перше число менше наступного за допомогою варіативної функції. Результат зберігається у result_1, result_2.

5.2 Опис логічної структури

Оголошуємо головну функцію main(), у котрій викликаємо функцію pairs. Викликаючи функцію задаємо кількість чисел у послідовності та саму послідовність. pairs визначає кількість пар сусідніх чисел, де перше число менше наступного.

5.3 Головна функція main()

Схема алгоритму подана на рисунку 8.

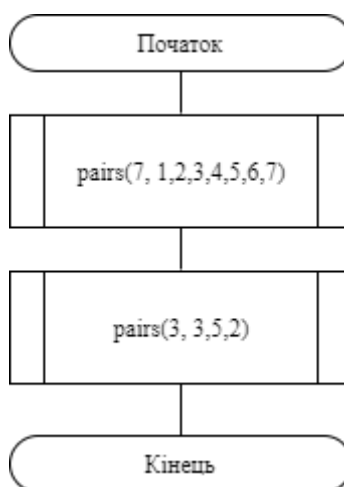


Рисунок 7 – Схема алгоритму головної функції main().

5.4 Функція `pairs`

Визначає кількість слів, розділених пробілами у заданому рядку. Схема алгоритму на рисунку 5.

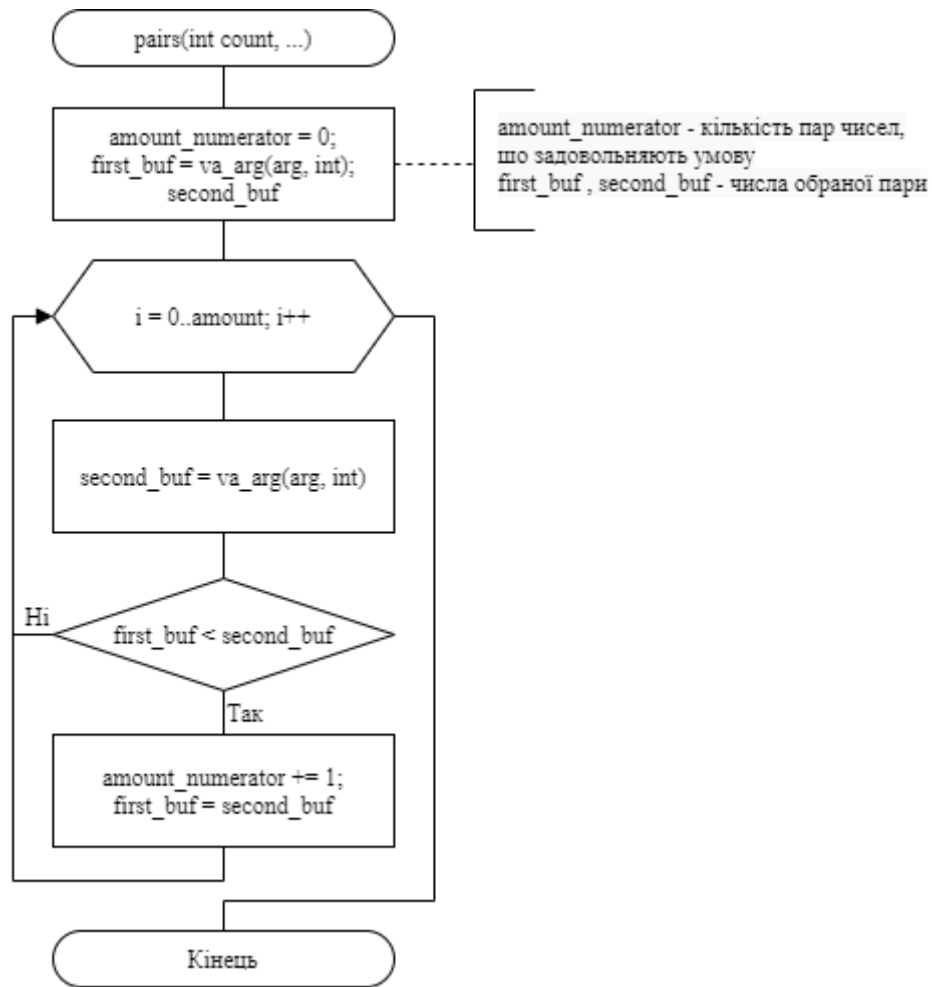


Рисунок 8 – Схема алгоритму функції wpum().

5.5 Результат виконання програми.

Використавши debug можна перевірити програму на дієздатність. Результат на рисунку 9.

Variable	Value
result_1	6
result_2	1

Рисунок 9 – результат виконання програми.

6. Висновки

При виконанні лабораторної роботи було набуто практичного досвіду розробки документації формату markdown.