

# **Лабораторна робота №10. Вступ до документації**

## **1 Вимоги**

### **1.1 Розробник**

- Михневич Владислав Вікторович
- Студент 1-го курсу
- Групи КІТ-120а

### **1.2 Загальне завдання**

Розробити повноцінний звіт для лабораторної роботи функції

### **1.3 Індивідуальні завдання**

- За допомогою функцій отримати корінь заданого числа.
- За допомогою функцій перемножити матрицю саму на себе.
- За допомогою варіативної функції визначити, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного.

## **2 Описи програм**

### **2.1 Програма main\_1.c**

#### **2.1.1 Функціональне призначення**

Програма добуває квадратний корінь з числа за допомогою функції `result_root()`. Результат зберігається у змінній `root`. Демонстрація результату передбачає покрокове виконання програми.

#### **2.1.2 Опис логічної структури програми**

- Для визначення квадратного кореня з числа викликаємо функцію `result_root()`, яка приймає параметр: число `n` з якого буде добуватись квадратний корінь. Функція збільшує значення параметру `root` на 0.01 поки значення `root * root` не буде дорівнювати заданому `n`.

- Головна функція `main()`. Задає випадкове число від 0 до 224. Викликає функцію `result_root`. Функція, що добуває корінь з числа `result_root`. Добуває квадратний корінь з числа. Параметри: `n` - число, з якого потрібно добути корінь; `root` - добутий корінь з заданого числа. Функція повертає `root`. Схема алгоритму функції (див. рис. 1).

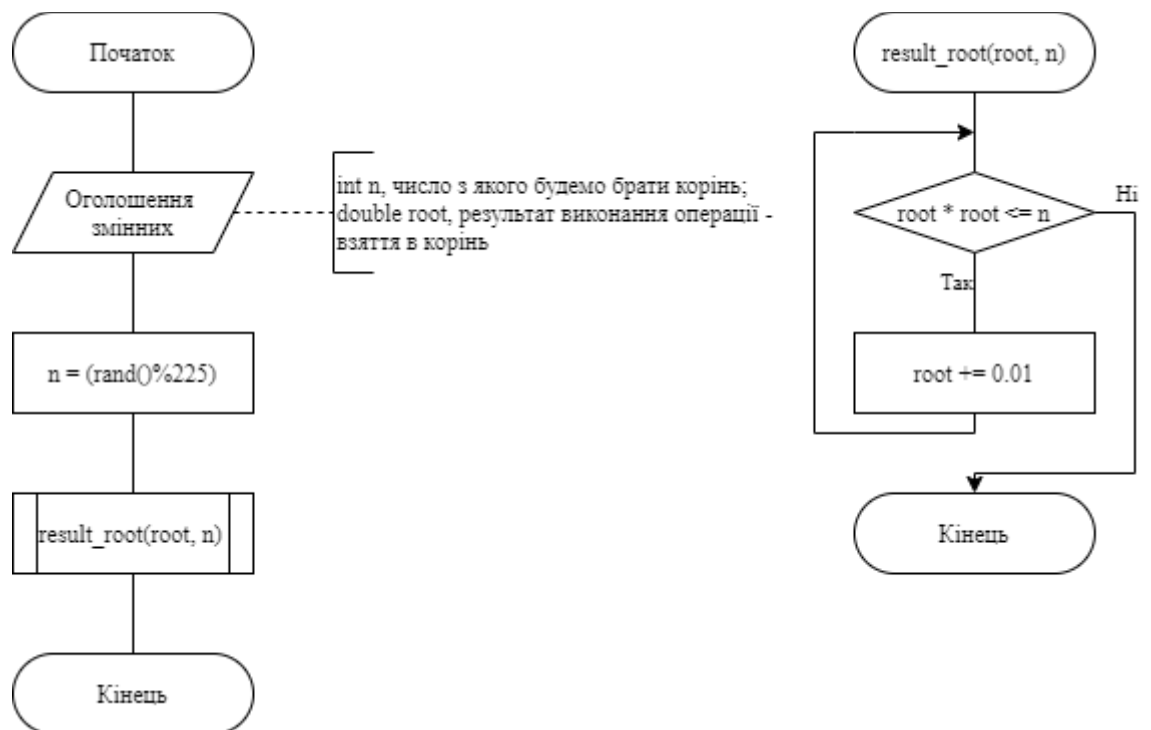


Рисунок 1 - Схема алгоритму функцій, завдання `main_1.c`

### 2.1.3 Важливі фрагменти програми

- Генерація випадкового числа.

```

srand(time(NULL));
n = (rand() % 225);

```

- Добування кореню за допомогою циклу.

```

while (root * root <= n) {
    root += 0.01;
}

```

### 2.1.4 Варіанти використання

З допомогою відладника ставимо точку зупинки на строках виклику функції (для визначення початкового стану ) та return 0, (див. рис. 2,3).

main\_1.c

1

#include <stdio.h>

2

#include <time.h>

3

#include <stdlib.h>

4

5

double result\_root(double root, int n);

6

7

int main() {

8

int n;

9

double root = 0;

10

11

srand(time(NULL));

12

n = (rand()% 225);

13

ID по	Кадр	Функція	Перемінна	Значення	Тип
1	0	main	Локальні перемінні		
			n	32767	int
			root	0	double

Рисунок 2 - Стан на початку функції в nemiver

19

double result\_root(double root, int n) {

20

21

while (root \* root <= n) {

22

root += 0.01;

23

}

24

25

return root;

26

Рисунок 3 - Стан при завершенні функції в nemiver

## 2.2 Програма main\_2.c

### 2.2.1 Функціональне призначення

Програма перемножує матрицю саму на себе за допомогою функції `multiplication()`. Демонстрація результату передбачає покрокове виконання програми.

### 2.2.2 Опис логічної структури програми

Для множення матриці викликаємо функцію `multiplication()`, яка за допомогою циклів перемножує матрицю саму на себе, за правилом множення матриць. Схема алгоритму функції (див. рис. 4).

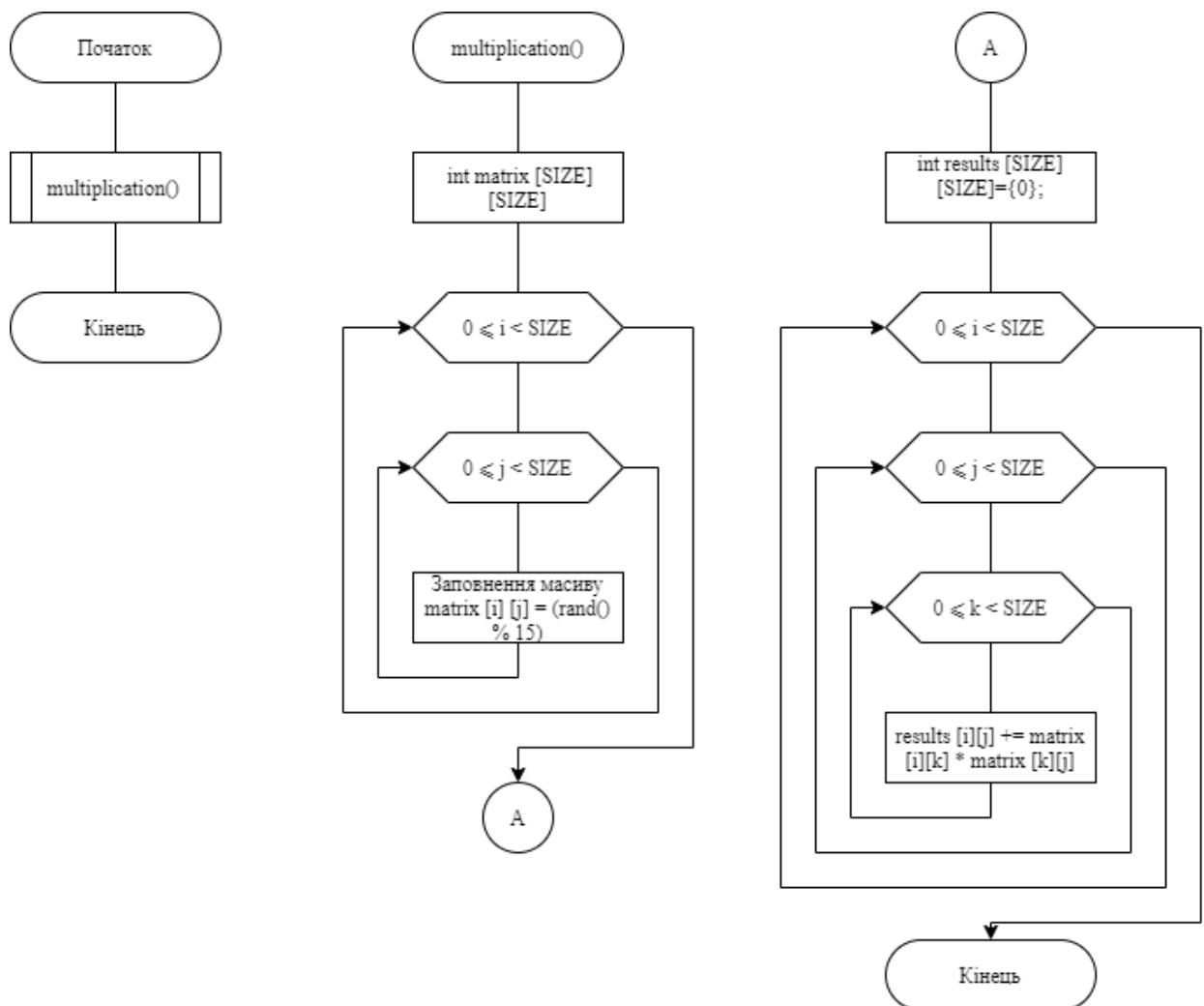


Рисунок 4 - Схема алгоритму функцій, завдання main\_2.c

## 2.2.3 Важливі фрагменти програми

- Підключення заголовочного `stdlib.h` та `time.h` для генерації випадкових чисел.
- Генерація випадкових чисел.

```
srand(time(NULL));  
  
for (int i = 0; i < SIZE; i++) {  
    for (int j = 0; j < SIZE; j++) {  
        matrix[i][j] = (rand() % 15);  
    }  
}
```

- Виклик функції для множення матриці `multiplication()`.
- Процес множення матриці саму на себе.

```
for (int i = 0; i < SIZE; i++) {  
    for (int j = 0; j < SIZE; j++) {  
        for (int k = 0; k < SIZE; k++) {  
            results[i][j] += matrix[i][k] * matrix[k][j];  
        }  
    }  
}
```

## 2.2.4 Варіанти використання

Для демонстрації результатів використовується покрокове виконання програми та інші засоби налагодження відладника `nemiver`, (див. рис. 5).

	1	main	matrix [3]			int [3][3]
			0	1	2	
			0	10	2	int
			1	2	3	int
			2	3	4	int
			0	6	11	int
			1	11	2	int
			2	4	9	int
			0	11	2	int
			1	2	9	int
			2	9	2	int

	▼ results	[3]	int [3][3]
	▼ 0	[3]	int [3]
	0	145	int
	1	48	int
	2	65	int
	▼ 1	[3]	int [3]
	0	170	int
	1	141	int
	2	98	int
	▼ 2	[3]	int [3]
	0	221	int
	1	62	int
	2	122	int

Рисунок 5 - Значення змінних в nemiver

## 2.3 Програма main\_3.c

### 2.3.1 Функціональне призначення

Програма, що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного за допомогою варіативної функції search.

### 2.3.2 Опис логічної структури програми

Для визначення кількості пар чисел викликаємо функцію search(int numbers, ...), яка приймає параметр: число numbers - кількість чисел в послідовності, та саму послідовність. Функція перевіряє кожен елемент ряду із усіма наступними, якщо елемент менший за один із наступних елементів то змінна result збільшується на один. Головна функція main(). Викликає функцію search, (див. рис. 6).

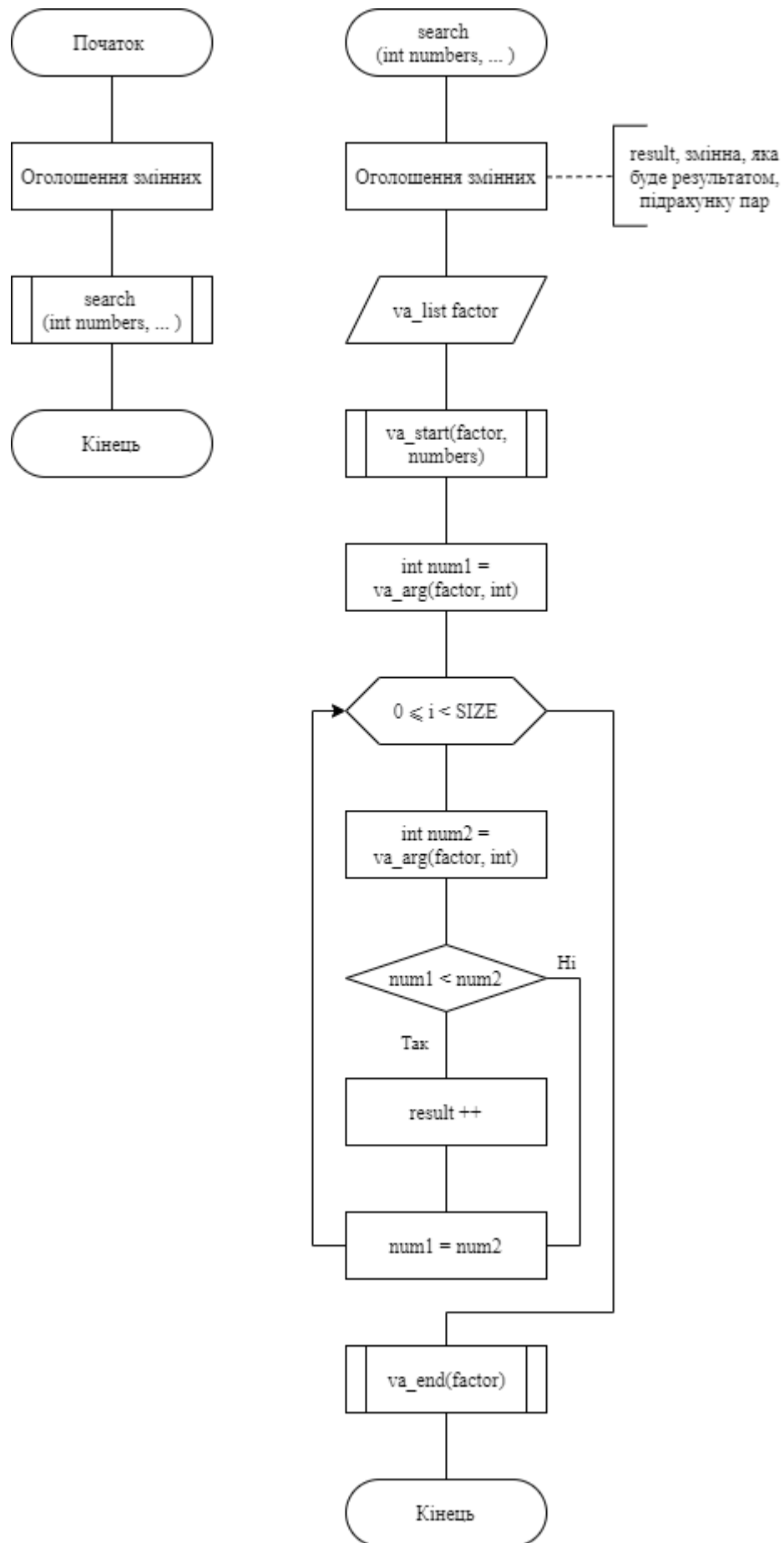


Рисунок 6 - Схема алгоритму функцій, завдання main\_3.c

### 2.3.3 Важливі фрагменти програми

- Підключення заголовочного файлу `<stdarg.h>` для роботи з варіативними функціями.
- Виклик варіативної функції та генерація випадкових чисел.

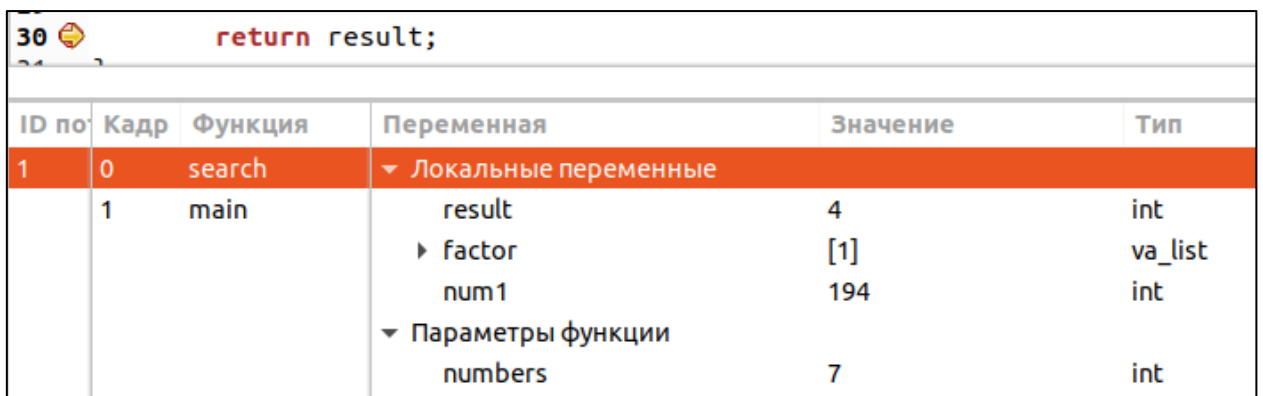
```
int result = search(SIZE, rand() % 15, rand() % 15, rand() % 15, rand() % 15, rand() % 15, rand() % 15, rand() % 15);
```

- Процес знаходження кількості пар чисел.

```
for (int i = 0; i < SIZE; i++){  
  
    int num2 = va_arg(factor, int);  
  
    if (num1 < num2) {  
  
        result ++;  
  
    }  
  
    num1 = num2;
```

### 2.3.4 Варіанти використання

Для демонстрації результатів використовується покрокове виконання програми та інші засоби налагодження відладчика `nemiver`, (див. рис. 7).



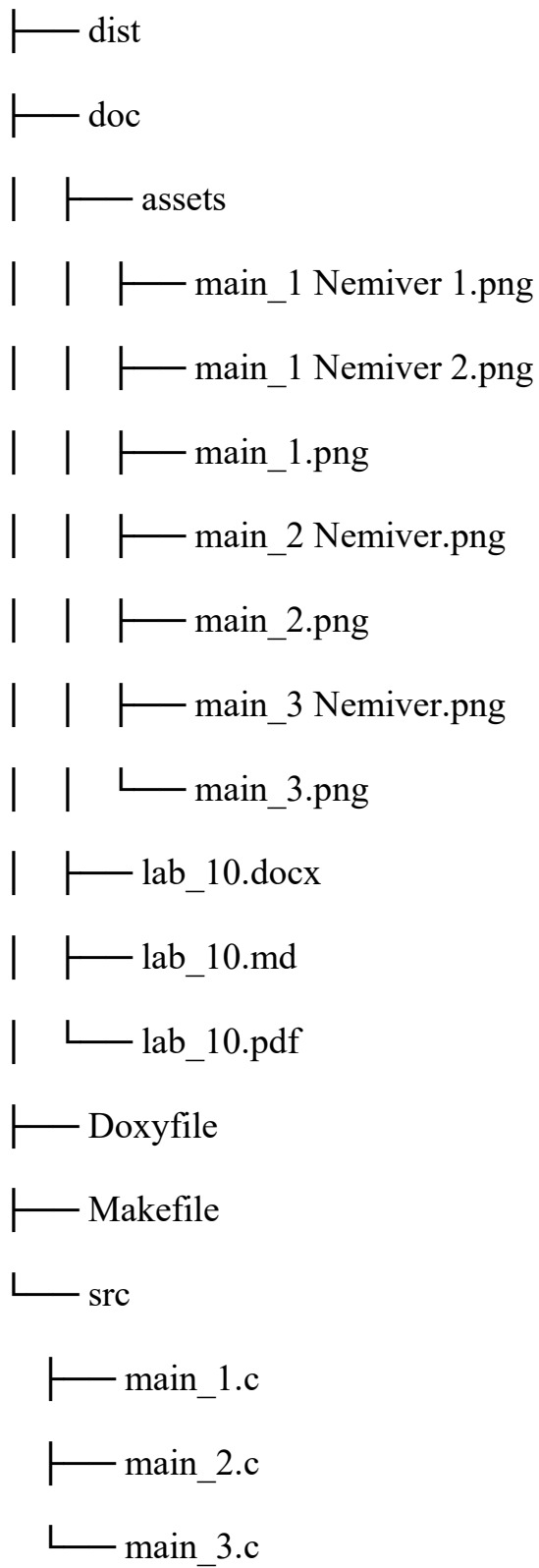
The screenshot shows the Nemiver debugger interface. At the top, a code editor displays the line `return result;` at line 30. Below the editor is a table with columns: ID по, Кадр, Функция, Переменная, Значение, and Тип. The table shows the state of the `search` function at frame 0. The `result` variable has a value of 4. The `factor` variable is an array with one element, 194. The `numbers` parameter has a value of 7.

ID по	Кадр	Функция	Переменная	Значение	Тип
1	0	search	Локальные переменные		
	1	main	result	4	int
			factor	[1]	va_list
			num1	194	int
			Параметры функции		
			numbers	7	int

Рисунок 7 - Значення змінних в `nemiver`



### 3 Структура проекта



## **Висновки**

На цій лабораторній роботі ми навчилися розробляти звіт для лабораторних робіт, за допомогою Markdown та в doc форматі згідно ГОСТам.