НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Обчислювальної техніки та програмування»

Звіт з лабораторних робіт №7, 8, 9, 10

Тема: **«Функції та вступ до документації коду»**

Виконав:

Лавка Олексій, гр. КІТ-120в

Харків 2020

**Мета: переробити завдання з попередніх лабораторних робіт через функції, розробити програму, використовуючи функцію з варіативними параметрами та написати до цих завдань doxygen.**

**Хід роботи**

**Перероблене завдання з 5 лабораторної роботи через функції**

**1.** Записуємо код

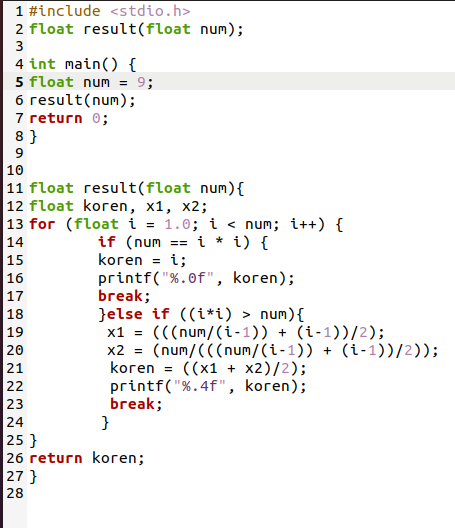


Рисунок 1 – код програми

**2.** Перевірка на правильність роботи програми через nemiver

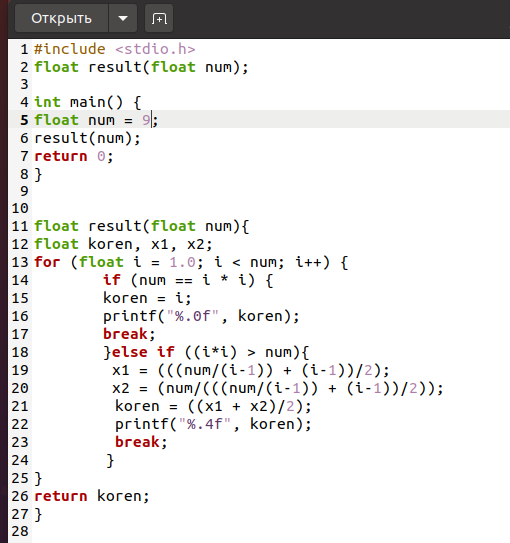


Рисунок 1 – код програми

**1.2. Перевірка на правильність роботи програми через nemiver**

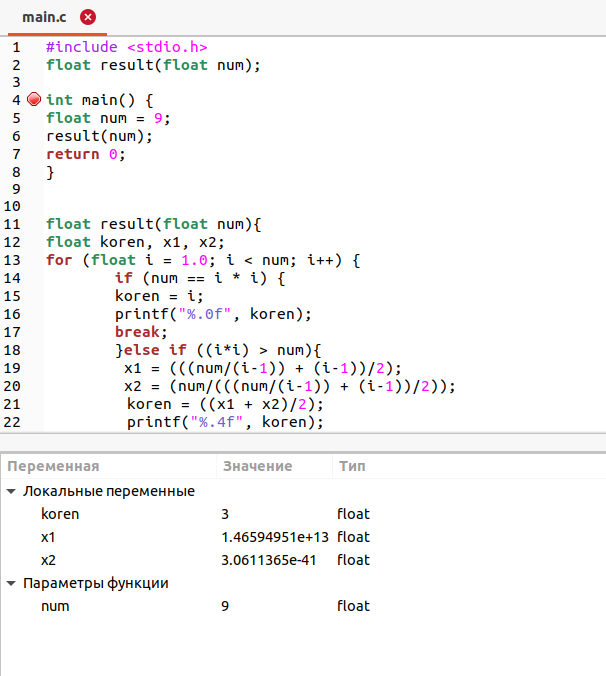


Рисунок 2 – nemiver

**3.** Блок-схема

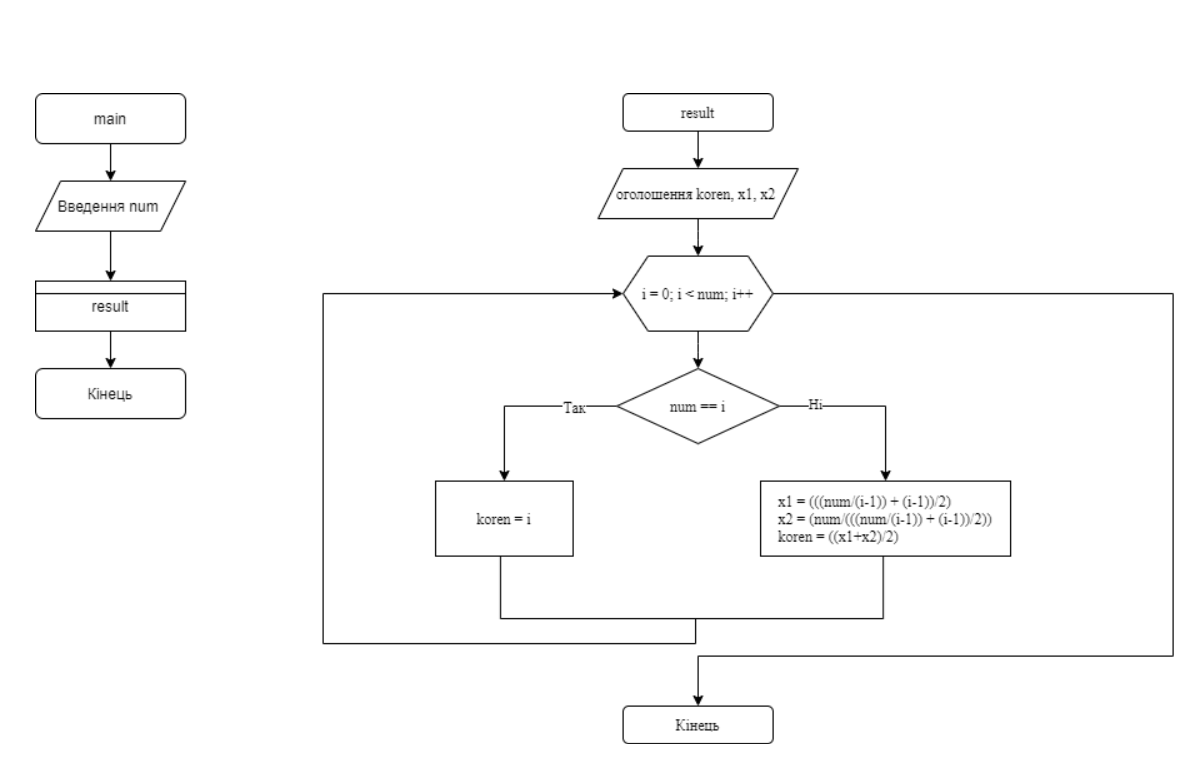


Рисунок 3 – блок-схема

**4.** Опис програми в doxygen

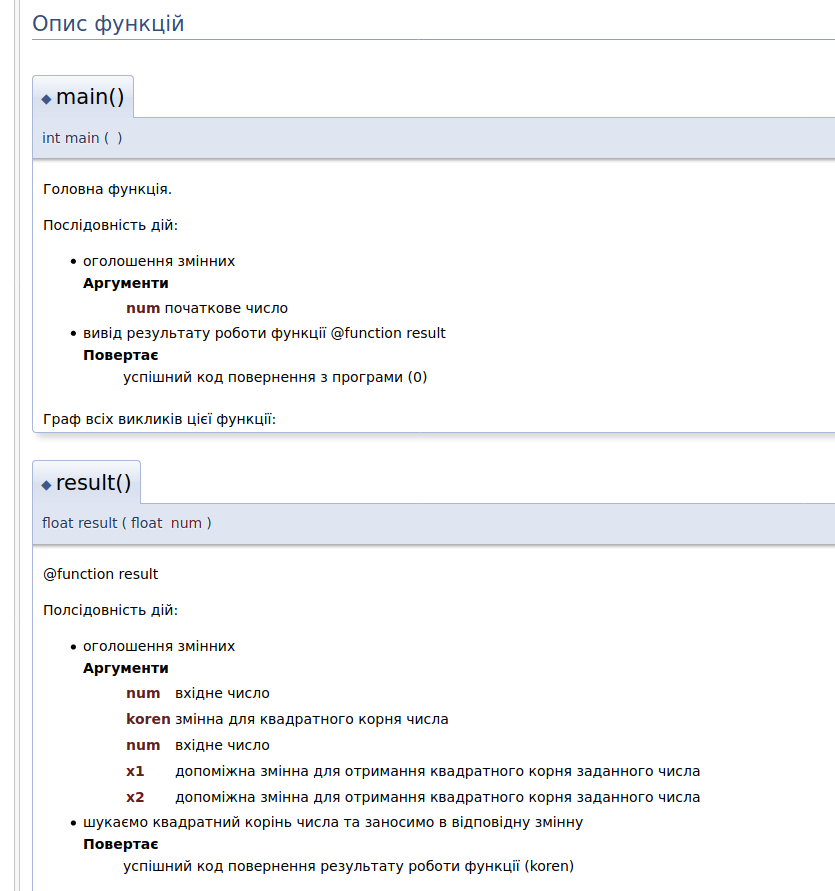


Рисунок 4 – doxygen

**Перероблюємо завдання з 6 лабораторної роботи через функції**

**1. Центрувати заданий рядок на площині з заданим заповнювачем**

**1.1.** Записуємо код



Рисунок 1.1 – код програми

**1.2.** Перевірка на правильність роботи програми через nemiver

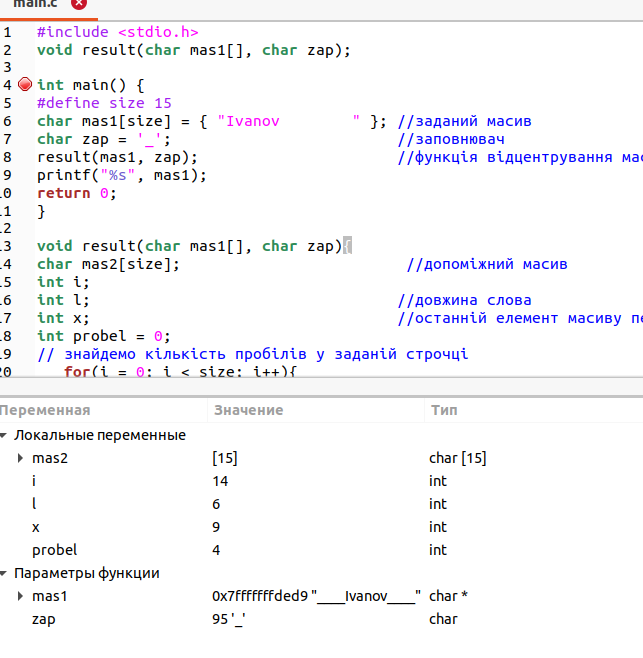


Рисунок 1.2 – nemiver

**1.3.** Блок-схема

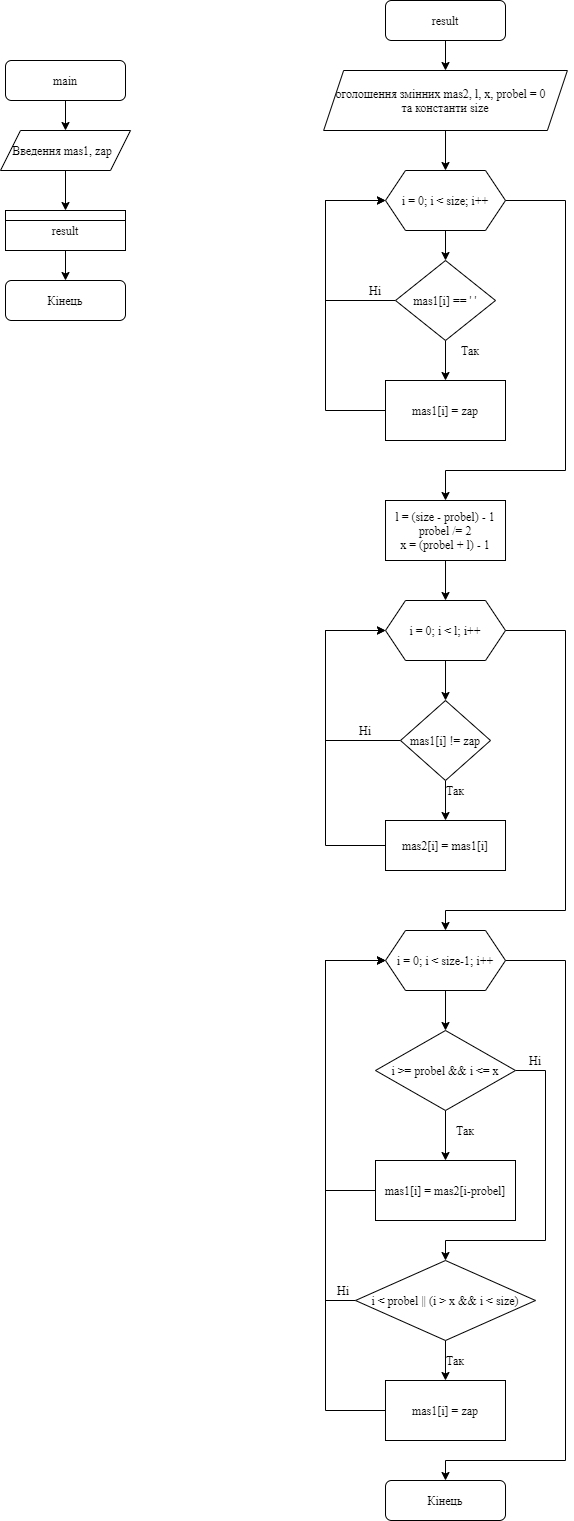


Рисунок 1.3 – блок-схема

**1.4.** Опис програми в doxygen

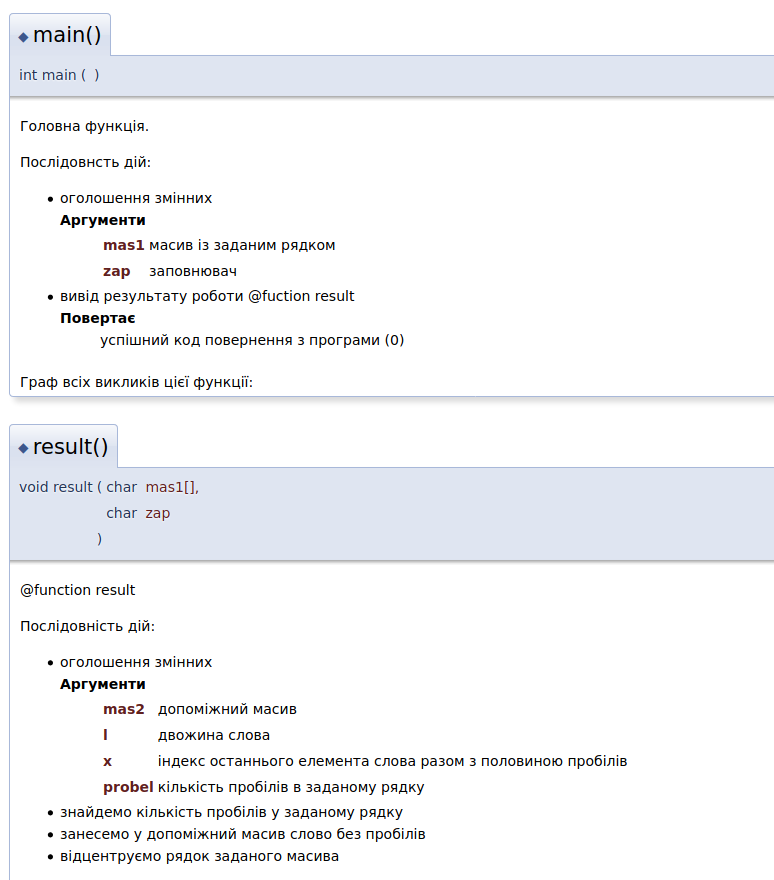


Рисунок 1.4 – doxygen

**2. Заповнити масив із заданої кількості елементів простими числами, що не повторюються. Кількість елементів може бути меньше за розмір масива**

**2.1.** Записуємо код

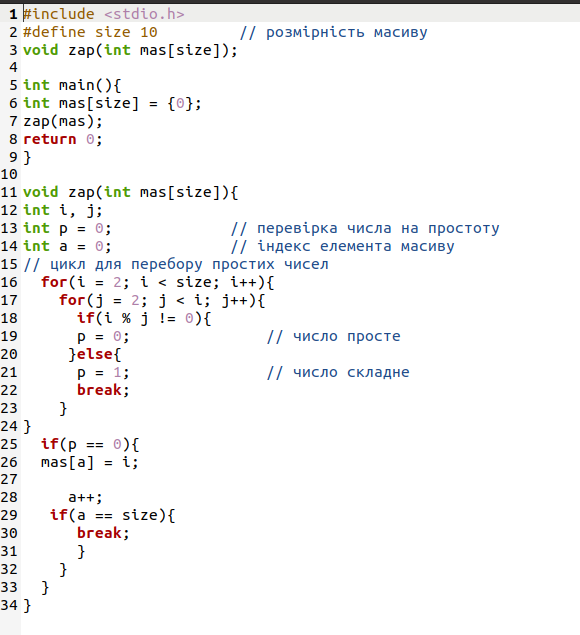


Рисунок 2.1 – код програми

**2.2.** Перевірка на правильність роботи програми через nemiver

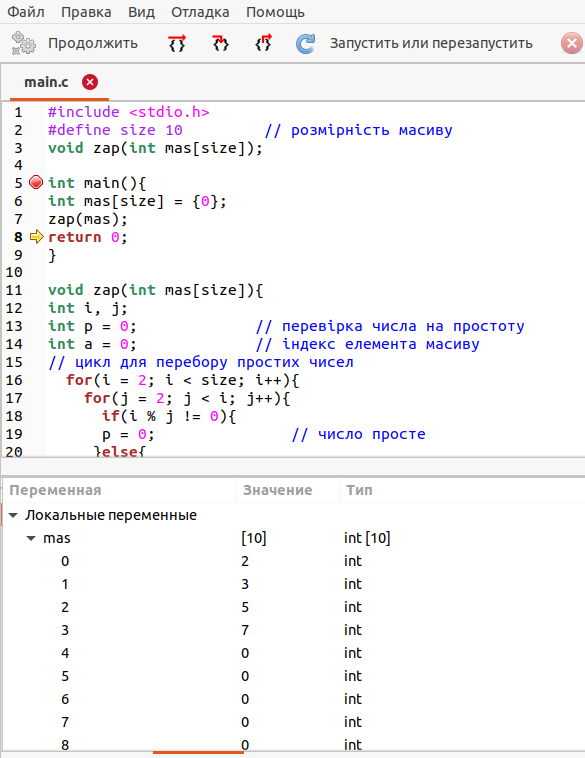


Рисунок 2.2 – nemiver

**2.3.** Блок-схема

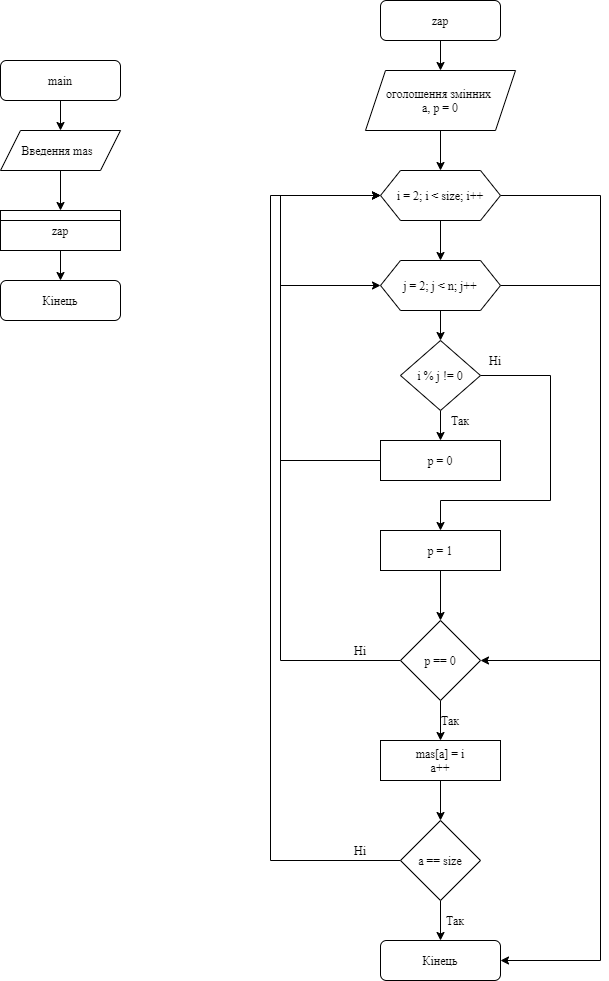
****

Рисунок 2.3 – блок-схема

**2.4.** Опис програми в doxygen

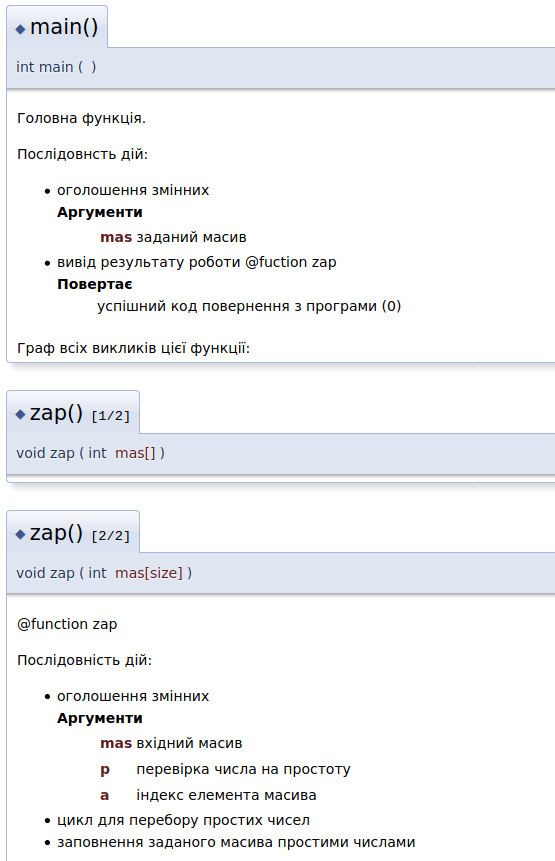


Рисунок 2.4 – doxygen

**3. Перетворити число в рядок**

**3.1** Записуємо код

#include <stdio.h>

#define size 100

void zapis(int num);

int main(){

int num = 2142;

zapis(num);

return 0;

}

void zapis(int num){

char x[size];

int n1, n2, n3, n4;

if(num > 9999){ //досить велике число

return 0;

//перевірка на нуль

}else if(num == 0){

x[0] = 'z';

x[1] = 'e';

x[2] = 'r';

x[3] = 'o';

printf("%s", x);

}else{ //якщо число не дорівнює нулю, тоді:

//задаєм масиви запису цифр словами

const char mas1[10][10] = {"", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"};

const char mas2[10][10] = {"ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen", "seventeen", "eighteen", "nineteen"};

const char mas3[10][10] = {"", "", "twenty ", "thirty ", "fourty ", "fifty ", "sixty ", "seventy ", "eighty ", "ninety "};

const char mas4[10][14] = {"", "one hundred ", "two hundred ", "three hundred ", "four hundred ", "five hundred ", "six hundred ", "seven hundred ", "eight hundred ", "nine hundred "};

const char mas5[10][15] = {"", "one thousand ", "two thousand ", "three thousand ", "four thousand ", "five thousand ", "six thousand ", "seven thousand ", "eight thousand ", "nine thousand "};

for(int i = 0; i < 100; i++){

x[i] = 0;

}

//цикл для запису тисяч в результуючий масив

for(int i = 0; i <= num/1000; i++){

n1 = 0;

for(int j = 0; j < 15; j++){

x[n1++] = mas5[i][j];

}

}

n2 = n1;

//цикл для запису сотен в результуючий масив

for(int i = 0; i <= ((num%1000)/100); i++){

n1 = n2;

for(int j = 0; j < 14; j++){

x[n1++] = mas4[i][j];

}

}

n3 = n1;

//цикл для запису десятків в результуючий масив

if(((num%100)/10) == 1){ //якщо число десятків дорівнює 1

for(int i = 0; i <= num%10; i++){

n1 = n3;

for(int j = 0; j < 10; j++){

x[n1++] = mas2[i][j];

}

}

}else if(((num%100)/10 != 1)){ //якщо число десятків не дорівнює 1

for(int i = 0; i <= ((num%100)/10); i++){

n1 = n3;

for(int j = 0; j < 10; j++){

x[n1++] = mas3[i][j];

}

}

n4 = n1;

//цикл для запису одиниць в результуючий масив

for(int i = 0; i <= num%10; i++){

n1 = n4;

for(int j = 0; j < 10; j++){

x[n1++] = mas1[i][j];

}

}

}

for(int i = 0; i < n1; i++){

printf("%c", x[i]);

}

}

}

**3.2.** Перевірка програми на правильність роботи через



Рисунок 3.1 – перевірка програми

**3.3.** Блок-схема

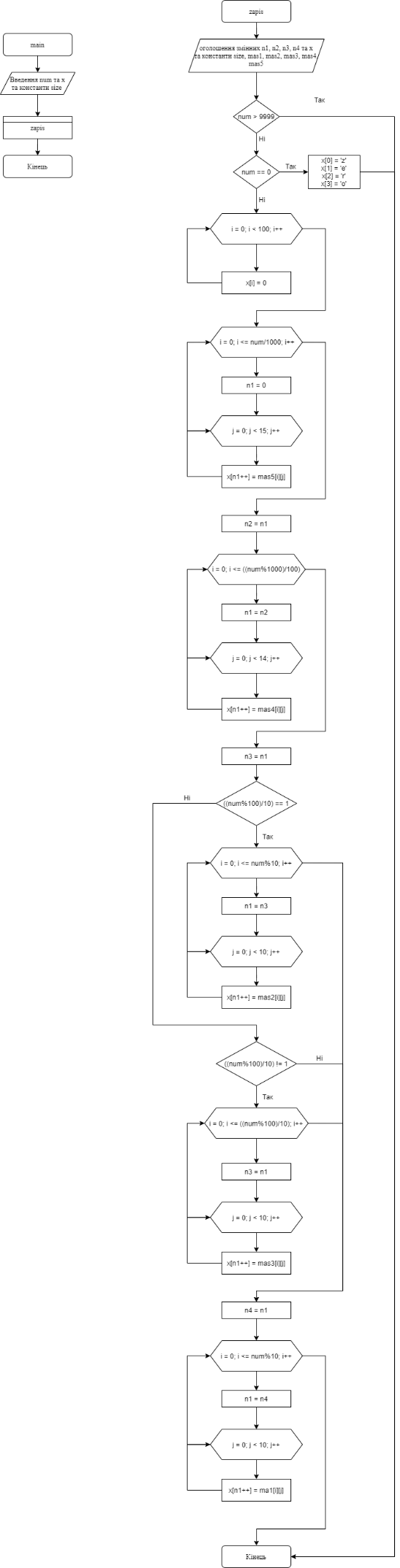


Рисунок 3.2 – блок-схема

**3.4.** Опис програми в doxygen

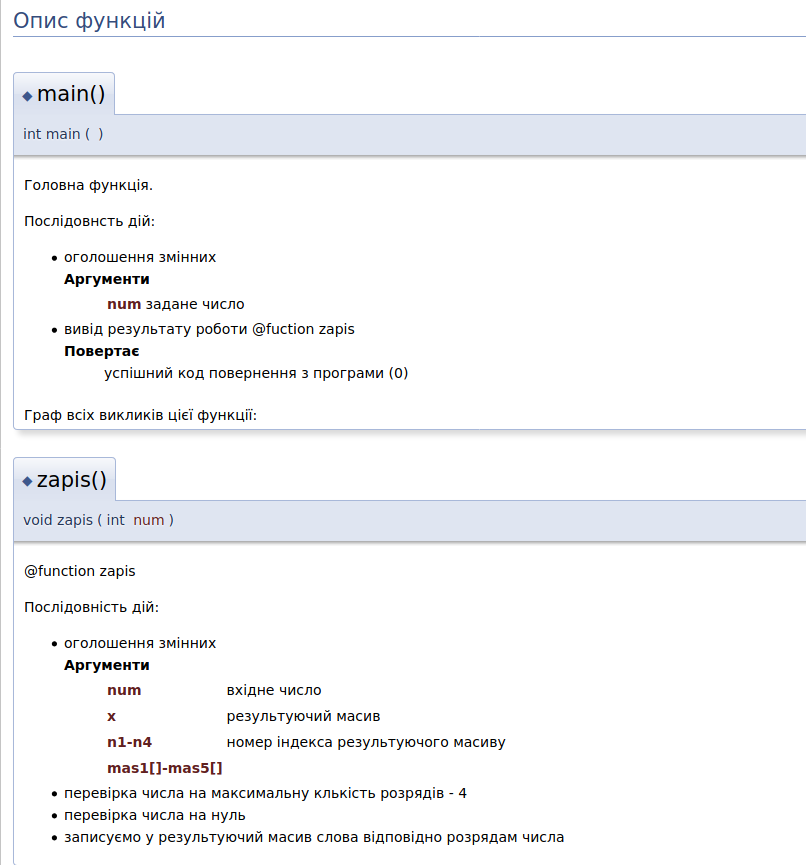


Рисунок 3.3 – doxeygen

**4. У заданому тексті знайти кількіть слів за умови, що між словами може бути будь-яка кількість пробілів.**

**4.1.** Записуємо код

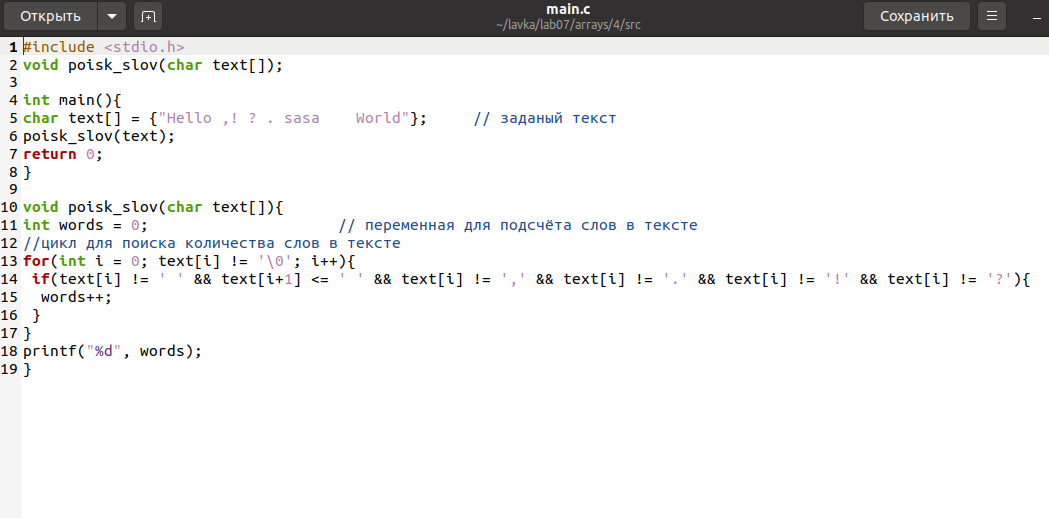


Рисунок 4.1 – код програми

**4.2.** Перевірка на правильність роботи програми через nemiver

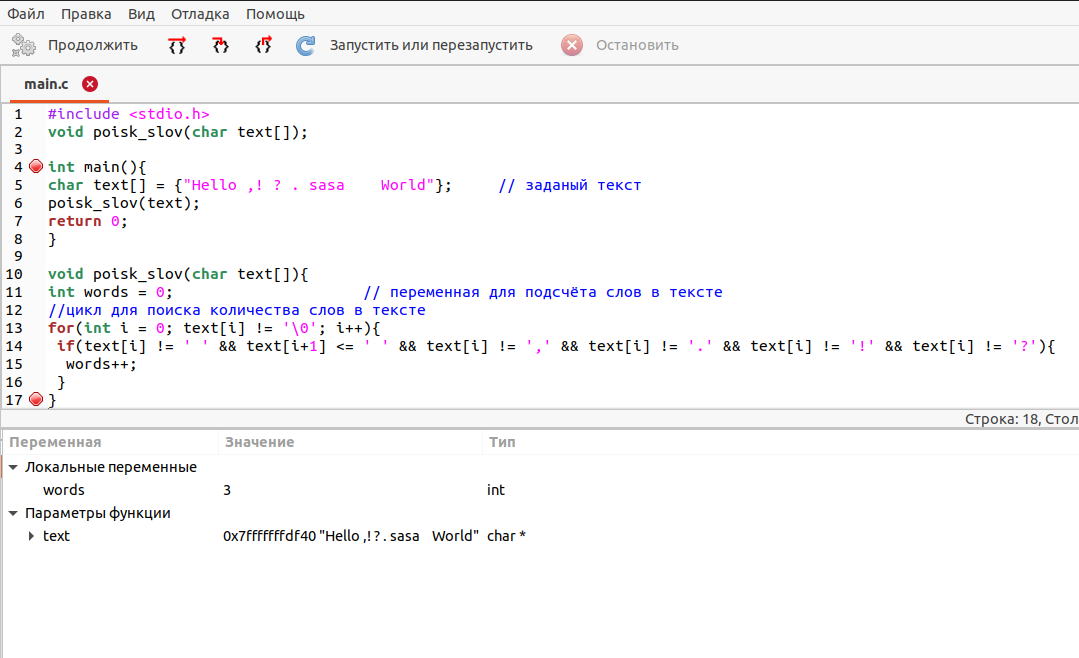


Рисунок 4.2 – nemiver

**4.3. Блок-схема**

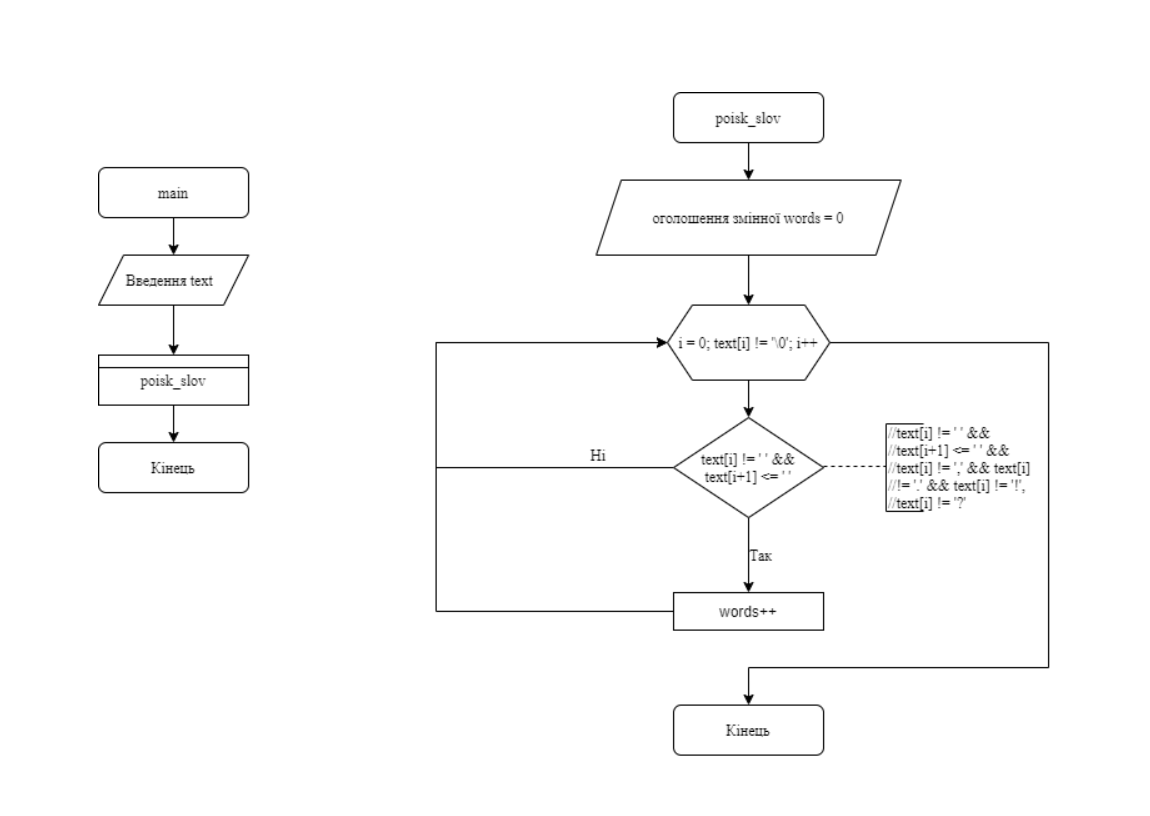


Рисунок 4.3 – блок-схема

**4.4.** Опис програми в doxygen

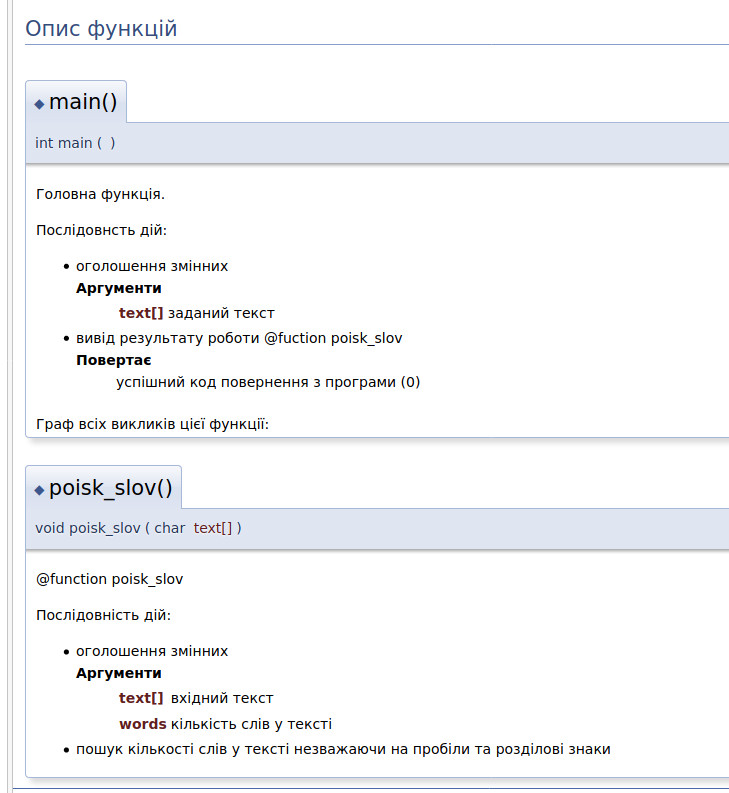


Рисунок 4.4 – doxygen

**5. Виконати циклічне зрушення елементів рядків масиву у напрямку зправа наліво.**

**5.1.** Записуємо код

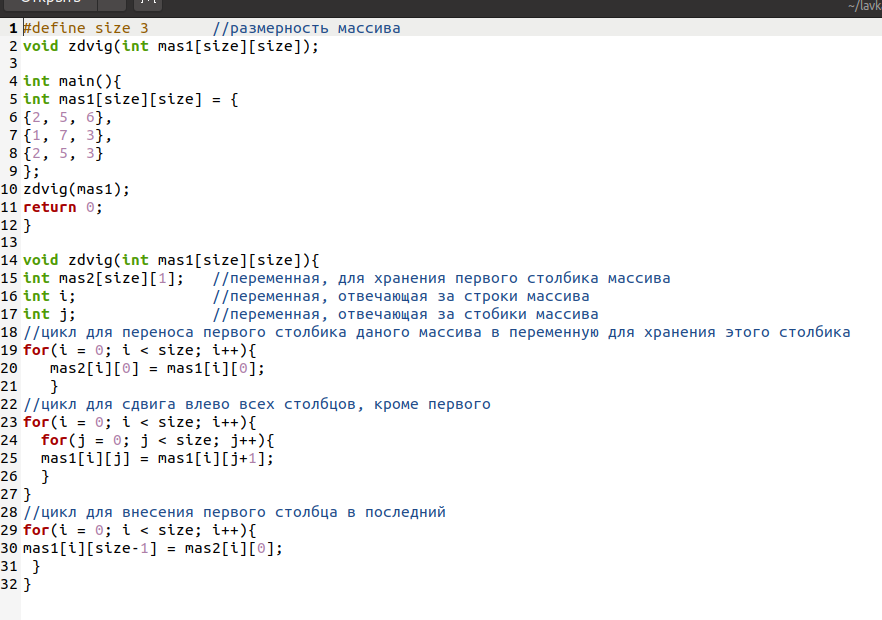


Рисунок 5.4 – код програми

**5.2.** Перевірка на правильність роботи програми через nemiver

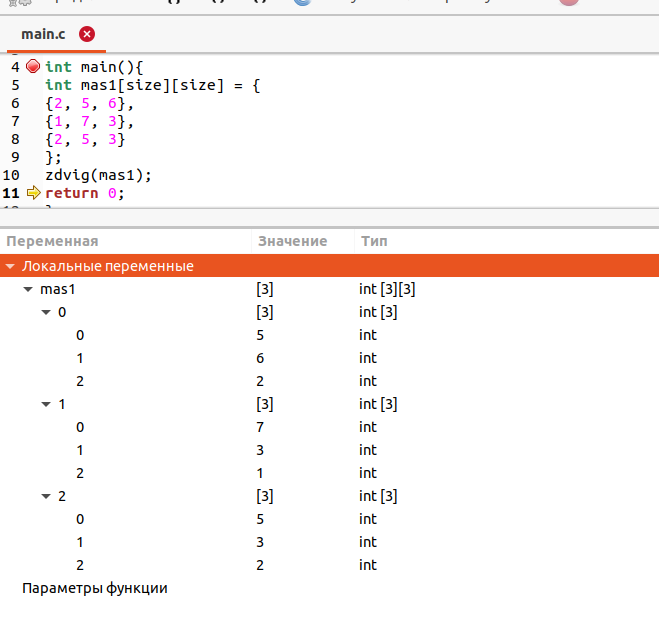


Рисунок 5.2 – nemiver

**5.3.** Блок-схема

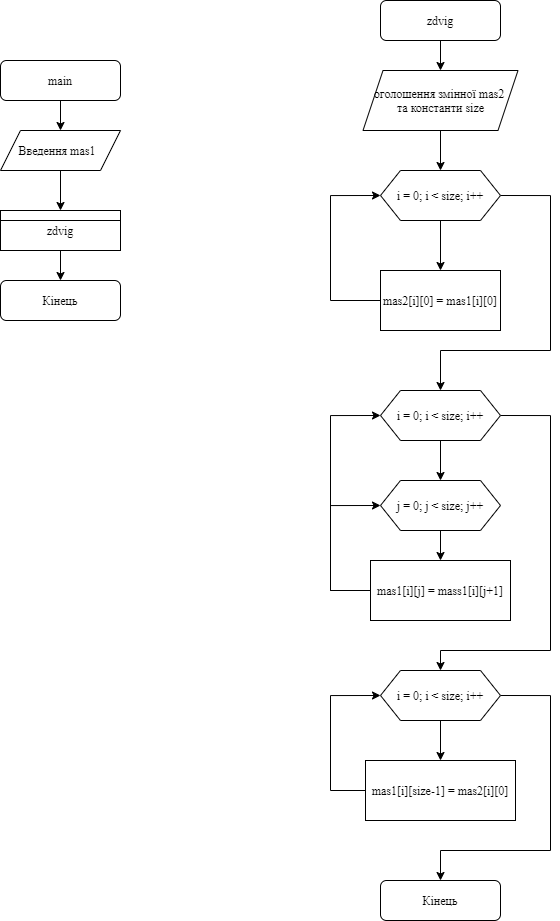


Рисунок 5.3 – блок-схема

**5.4.** Опис програми в doxygen

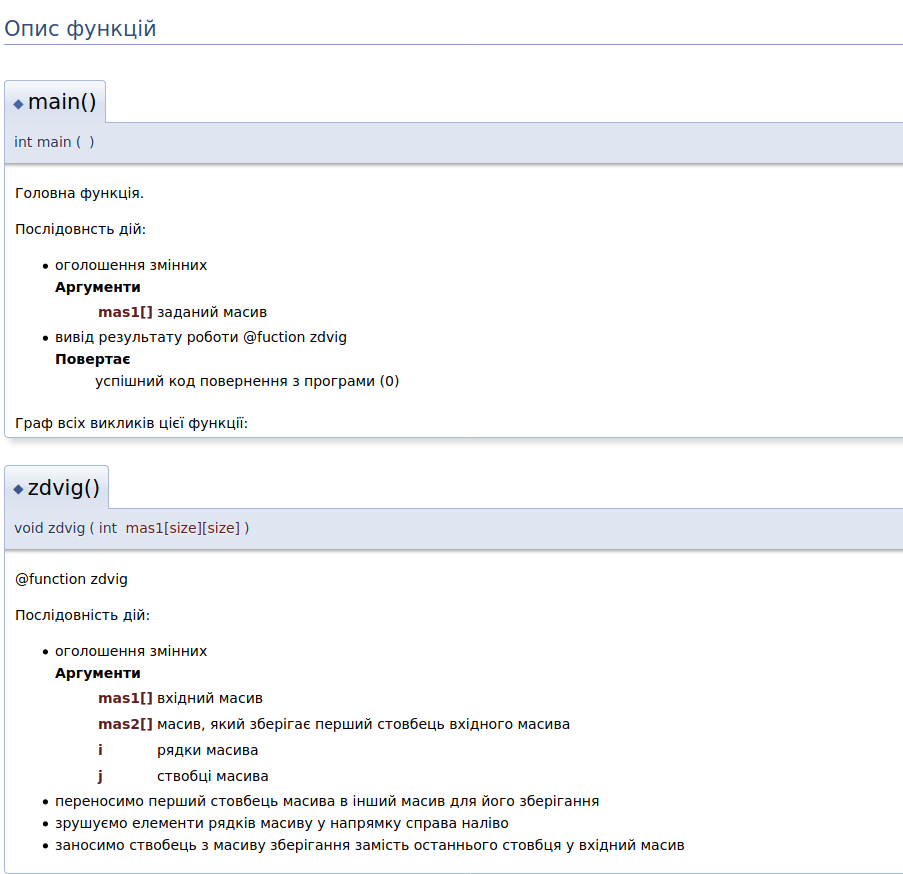


Рисунок 5.4 – doxygen

**6. Помножити матрицю саму на себе.**

**6.1.** Записуємо код

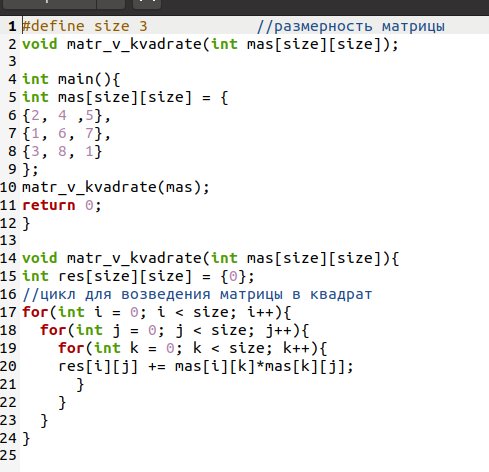


Рисунок 6.1 – код програми

**6.2.** Перевірка на правлильність роботи програми через nemiver

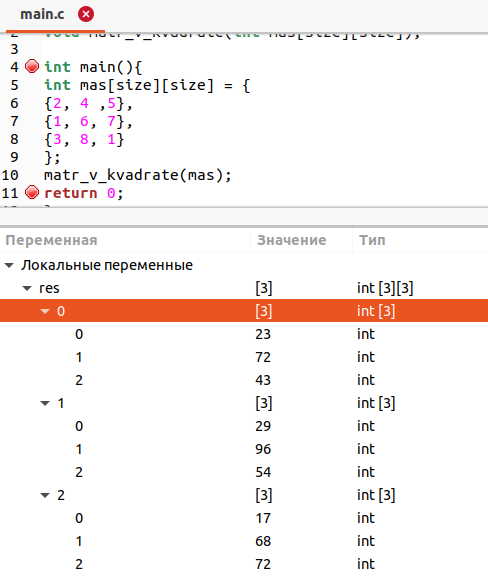


Рисунок 6.2 – nemiver

**6.3.** Блок-схема

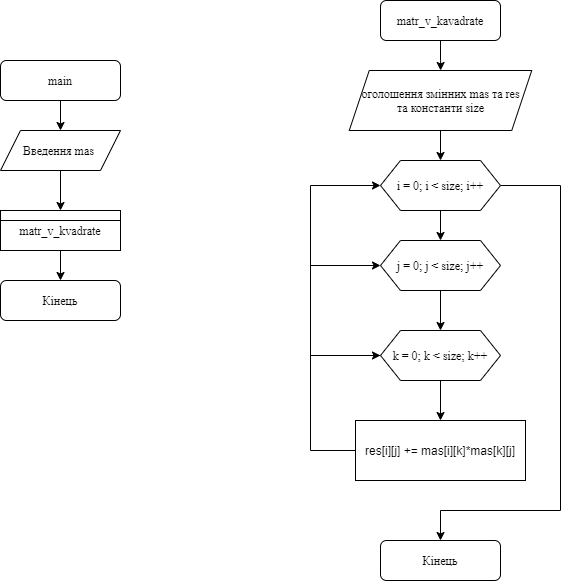


Рисунок 6.3 – блок-схема

**6.4.** Опис програми в doxygen

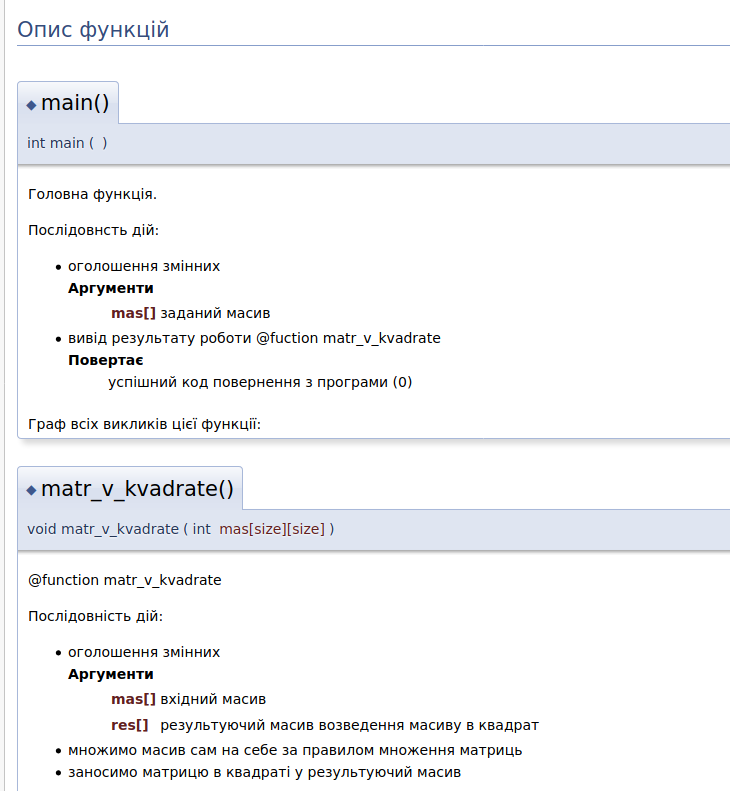


Рисунок 6.4 – doxygen

**Виконати завдання з лабораторної роботи №7 на відмінно: створити программу, використовуючи функцію з варіативними параметрами, яка буду рахувати скільки черед заданих пар чисел таких пар, у котрих перше число меньше наступного.**

**7.1.** Записуємо код

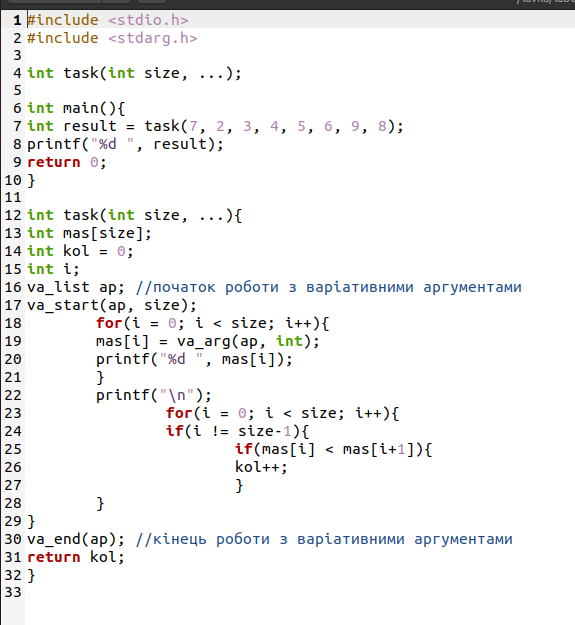


Рисунок 7.1 – код програми

**7.2.** Перевірка на правильність роботи програми через nemiver

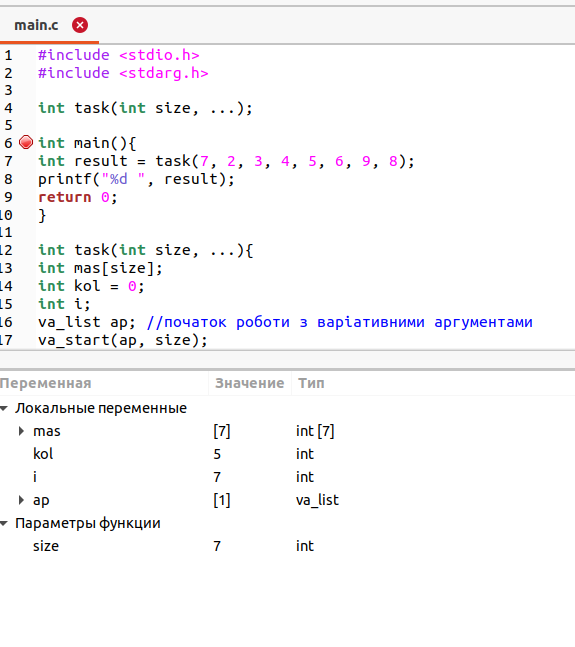


Рисунок 7.2 – nemiver

**7.3.** Блок-схема

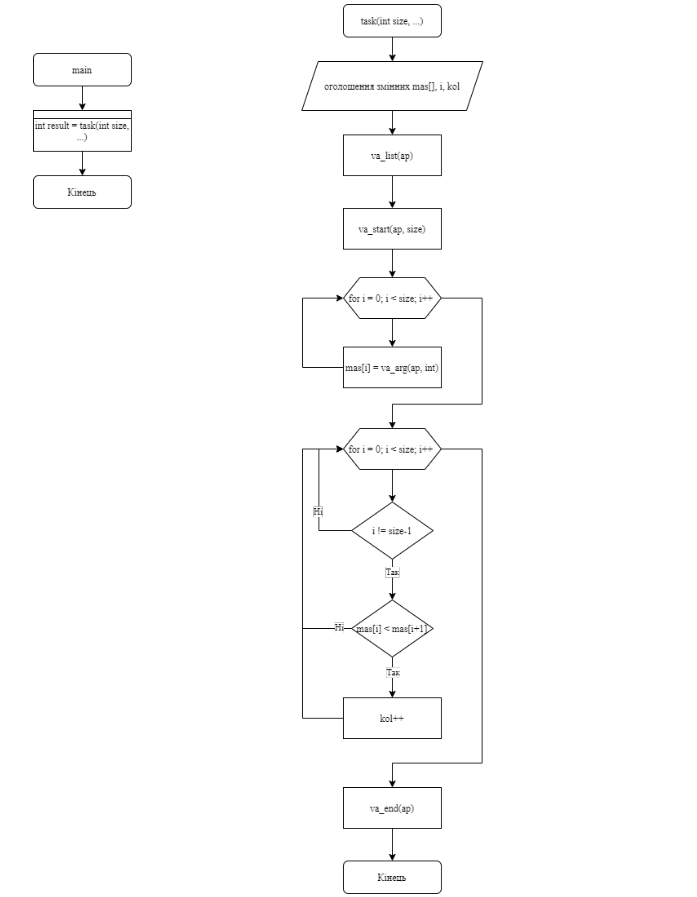


Рисунок 7.3 – блок-схема

**7.4.** Опис програми в doxygen

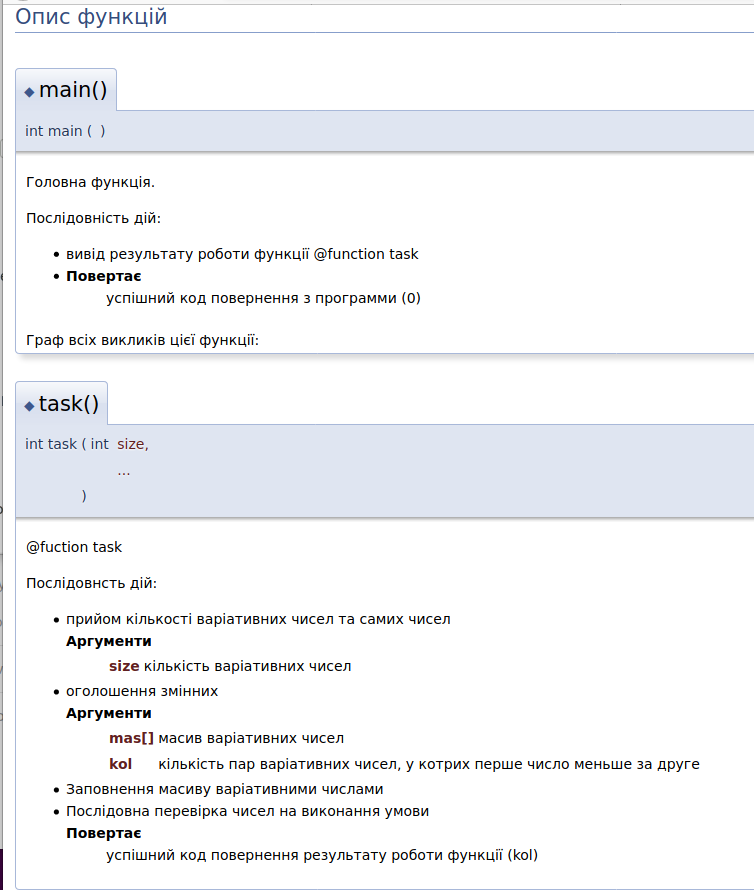


Рисунок 7.4 – doxygen

**Висновок:** у цьому звіті я описав одразу 4 лабораторні роботи. У лабораторній роботі №7 я ознайомився з функціями та варіативними параметрами. У лабораторній роботі №8 я створив блок-схеми алгоритмів для кожного завдання. У лабораторній роботі №9 я навчився писати doxygen. У лабораторній роботі №10 я навчився робити звіт у формі markdown.