

Лабораторна робота №9 Документація коду

1 Вимоги

1.1 Розробник

- Носов Микола
- студент групи кіт 1206
- 8.12.2020

1.2 Мета:

Навчитися складати документацію до коду, а також автоматизувати генерацію цієї документації

1.3 Загальне завдання

Для лабораторної роботи “Функції” необхідно додати можливість генерації Doxygen документації.

Починаючи з цієї роботи Makefile повинен генерувати Doxygen документацію (наприклад, додав ціль `format`). У звіті секції опису даних повинні бути скопійовані з результатів Doxygen документації.

Структура проекту лабораторної роботи (додаткові файли до існуючих, що були створені для л/р “Функції”):

```
.
├─ labXX/
│   └─ Doxyfile
│       └─ dist/
│           └─ html/
```

2 Опис програми

2.1 Функціональне призначення

Програма призначена для:

- знаходження НОД 2 чисел;
- перевірки на білета на “щастливість”;
- перевірки числа, і визначення, чи є воно простим;
- перевірки числа, і визначення, чи є ідеальним;
- знаххождення квадратного кореня числа;
- центрування слів у строчці;

2.2 Логічна структура проекту

Структура проекту (див рис. 1)

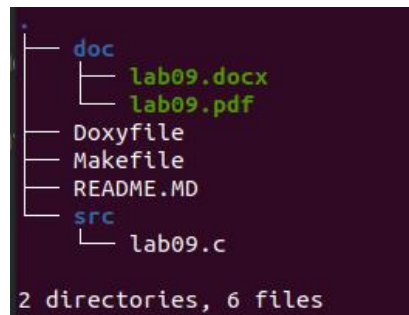


Рисунок 1 - робота утиліти tree

Список функцій програми (див рис. 2)

- centrovka() : lab10.c
- couples() : lab10.c
- ideal_digit() : lab10.c
- lucky_ticket() : lab10.c
- main() : lab10.c
- nod1() : lab10.c
- prime_digit() : lab10.c
- sqrt1() : lab10.c

Рисунок 2 - список функцій та посилання на них у документації

1. Функція знаходження НОД (див рис. 3)

```
5 int nod1( int a, int b )
6 {
7     int nod;
```

Рисунок 3 - фрагмент коду

Призначення :

-Знаходження найбільшого спільного дільника ;

Аргументи :

-2 числа;

Опис роботи :

-визначення, яке число число є більшим;

-ділення більшого числа на менше по %, доки вони $\neq 0$;

-сумуємо 2 числа;

Опис у doxygen:(див рис. 4)

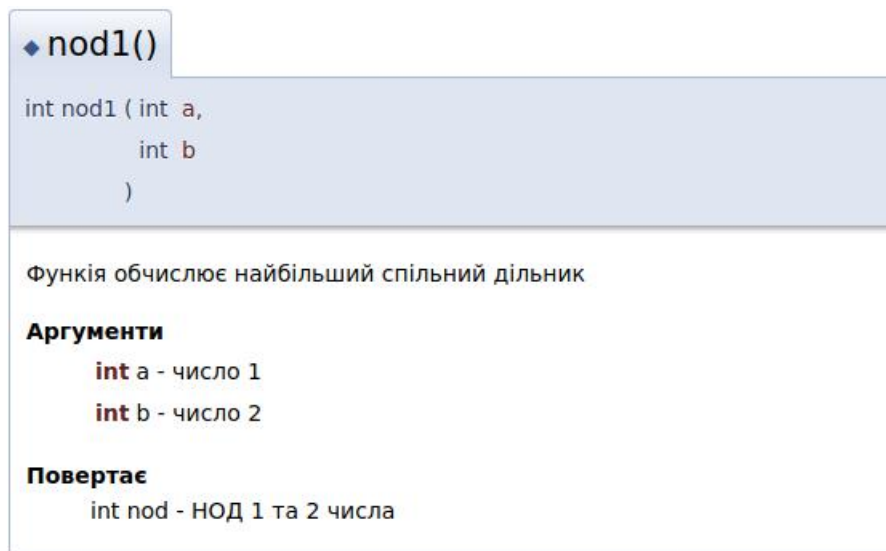


Рисунок 4 - фрагмент документації

2. Функція визначення щасливого білетика (див рис. 5)

```
34 int lucky_ticket(int n)
35 {
36     int x2 = 0;
37     int x3 = 0;
38     int j = 0;
```

Рисунок 5 - фрагмент коду

Призначення :

-Визначення, чи є білетик щасливим;

Опис роботи :

-розбиття 6-ти значного числа на 6 цифр;

-сума перших трьох чисел;

-сума чисел, що залишилися;

-якщо суми однокові - білетик щасливий;

Опис у doxygen:(див рис 6)



Рисунок 6 - фрагмент документації

Схема алгоритму (див лаб 8) :

3. Функція визначення простого числа (див рис. 8)

```
74 int prime_digit(int a12)
75 {
76     int result;
```

Рисунок 8 - фрагмент коду

Призначення :

-Визначення, чи є число простим;

Аргументи :

-Число

Опис роботи :

-Ділити число на всі числа від 1 до квадратного кореня заданого числа;

-Якщо число ділиться на ціло - число не просте, в іншому випадку -

просте;

Опис у doxygen:(див рис 9)

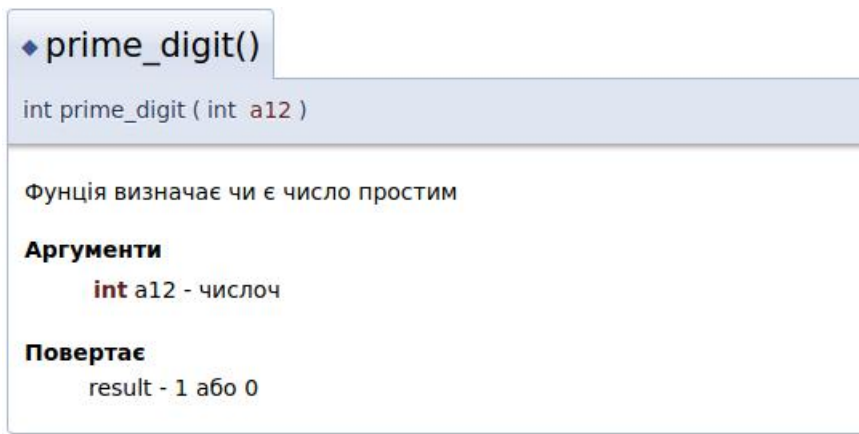


Рисунок 9 - фрагмент документації

4. Функція визначення ідеально числа (див рис.10)

```
int ideal_digit(int bi )  
{  
    int pi = 1;  
    int ci = 0;
```

Рисунок 10 - фрагмент коду

Призначення :

-Визначення, чи є число ідеальним;

Аргументи :

-Число;

Опис роботи :

-знаходження дільників числа;

-якщо сума дільників = самому числу - воно ідеальне;

Опис у doxygen:(див рис 11)

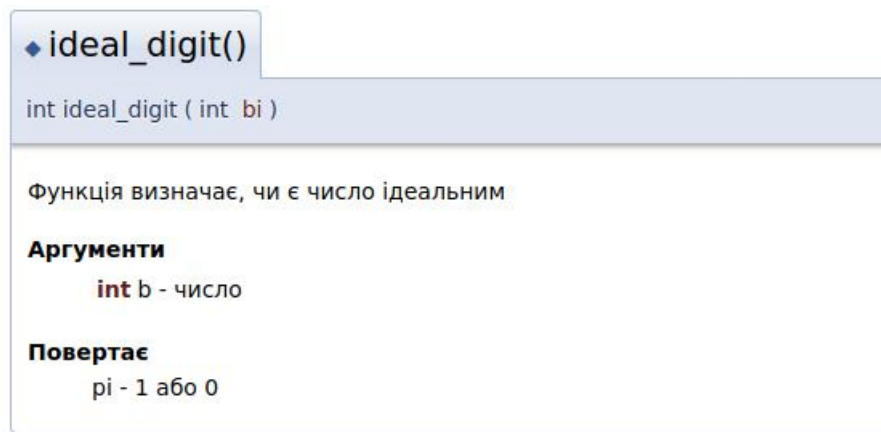


Рисунок 11 - фрагмент коду

5. Функція знаходження квадратного кореня (див рис 12)

```
float sqrt1(float chislo1)
{
    int chislo;
    float sqrt;
    float podkor_proiz;
    float chel_proiz;
    char s = '√';
    float digit = chislo1;
```

Рисунок 12 - фрагмент коду

Призначення :

-Знаходження квадратного кореня;

Аргументи :

-Число

Опис роботи :

-знаходження дільників числа;

-попарне винесення дільників числа із під знаку кореня;

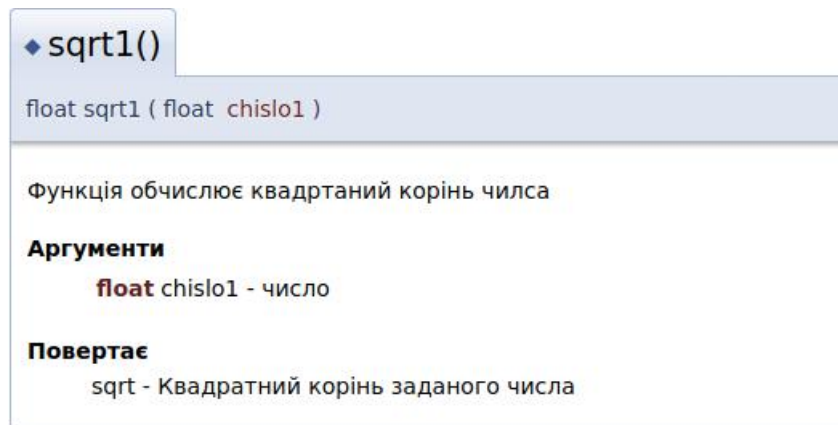
-знаходження кореня за теоремою дотичних Ньютона;

Отримаємо відповідь у такому вигляді (див рис 13):

$10\sqrt{2} \sim 14.142136$

Рисунок 13 - квадратний корінь з числа 200

Опис у doxygen:(див рис 14)



◆ sqrt1()

float sqrt1 (float chislo1)

Функція обчислює квадратний корінь числа

Аргументи

float chislo1 - число

Повертає

sqrt - Квадратний корінь заданого числа

Рисунок 14 - фрагмент документації

6. Функція центрування слова у рядку (див рис 15)

```
void centrovka(char mass[])
{
    int centr = 0;
    int shetBukv = 0;
    int firstChar;
```

Рисунок 15 - функція, що нічого не повертає

Призначення :

-Центрування слова або словосполучення у рядку;

Аргументи :

-Слово, що потребує цетрування;

Опис роботи :

-Знаходження першої букви;

-Підрахунок кількості букв;

-Розрахунок необхідної кількості символів центрування;

-Заповнення рядка буквами та символами “_”;

Отримаємо відповідь у такій формі (див рис 16)

__Ivan Bogov__

Рисунок 16 - результат роботи функції 6

Опис у doxygen:(див рис 17)

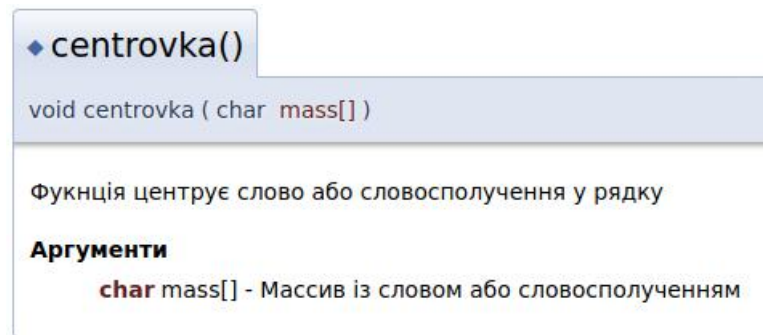


Рисунок 17 - фрагмент документації

Схема алгоритму (див рис лаб 8):

7. Функція з варіативною кількістю аргументів (див рис 21)

```
int couples(int var, ...)  
{  
    int count = 0;  
    int value = 0;  
    va_list vl;  
    va_start(vl, var);  
    int mass[var];
```

Рисунок 21 - фрагмент коду

Призначення :

-Знаходження кількості пар чисел, перше число якої менше за наступне;

Аргументи :

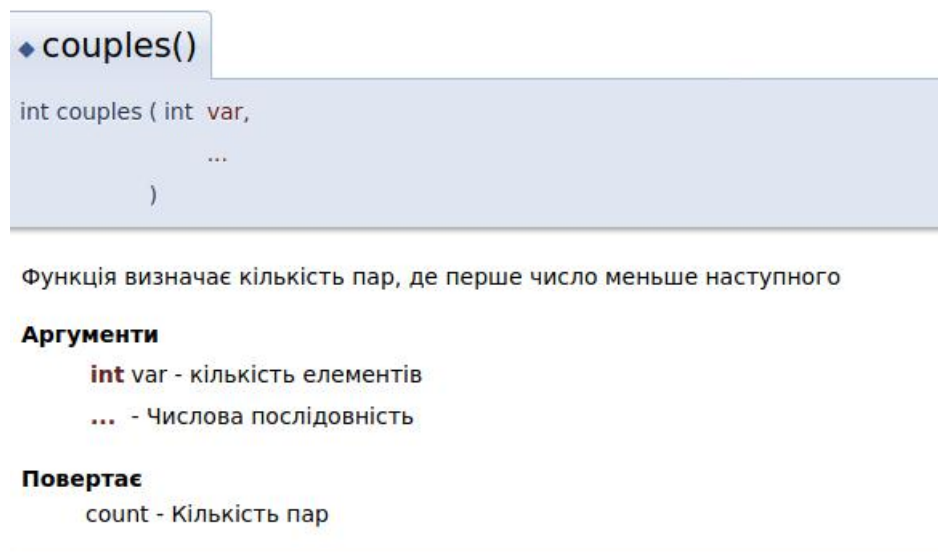
-Ряд чисел;

Опис роботи :

-Програма порівнює 2 числа, що стоять поруч;

-Якщо число менше наступного - образується пара;

Опис у doxygen:(див рис 22)



The screenshot shows the doxygen documentation for the `couples()` function. It includes the function signature `int couples (int var, ...)`. Below the signature, there is a description: "Функція визначає кількість пар, де перше число менше наступного". Then, under the heading "Аргументи", it lists the parameters: `int var` - кількість елементів and `...` - Числова послідовність. Finally, under the heading "Повертає", it states: `count` - Кількість пар.

```
◆ couples()  
int couples ( int var,  
             ...  
             )  
  
Функція визначає кількість пар, де перше число менше наступного  
  
Аргументи  
    int var - кількість елементів  
    ... - Числова послідовність  
  
Повертає  
    count - Кількість пар
```

Рисунок 22 - фрагмент документації

Схема алгоритму (див лаб 8):

3 Варіанти використання

Для детальнішого перегляду doxygen документації коду:

- клонуємо цей репозиторій с gitlab (git clone “url”)
- прописуємо команду doxygen Doxyfile
- переходимо у потрібну нам папку cd/dist/html
- запускаємо сторінку firefox (або інший браузер) index.html

Висновок:

Протягом даної лабораторної роботи ми навчилися генерувати doxygen документацію.