

Лабораторна робота №10 Вступ до документації проекту

1 Вимоги

1.1 Розробник

- Носов Микола
- студент групи кіт 1206
- 8.12.2020

1.2 Мета:

Навчитися писати повноцінні звіти у двох форматах doc та markdown (звіт стосується 7 лабораторної роботи)

1.3 Загальне завдання

Розробити повноцінний звіт для лабораторної роботи “Функції”, що присвячена функціям у двох форматів (+їх репрезентація у PDF форматі):

- *Markdown*
- *doc* формат, згідно ДСТУ

Детальну інформацію можна отримати за посиланнями у стосовних підрозділів розділу “Перелік літератури”

За складом, обидва звіту повинні повністю задовольняють вимогам оформлення звітів, що описаний у початку розділу лабораторних робіт першого семестру.

Починаючи з цієї роботи звіт має бути оформленим у Markdown стилі або у doc форматі згідно ДСТУ (за попередньою домовленістю з викладачем), згідно декларованим вимогам.

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

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

1. Функція знаходження НОД (див рис. 2)

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

Рисунок 2 - 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;

2. Функція визначення щасливого білетика (див рис. 3)

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

Рисунок 3 - фрагмент коду

Призначення :

-Визначення, чи є білетик щасливим;

Аргументи :

-6-ти значне число;

Опис роботи :

-розбиття 6-ти значного числа на 6 цифр;

- сума перших трьох чисел;
- сума чисел, що залишилися;
- якщо суми однокові - білетик щасливий;

Схема алгоритму (див рис 4) :

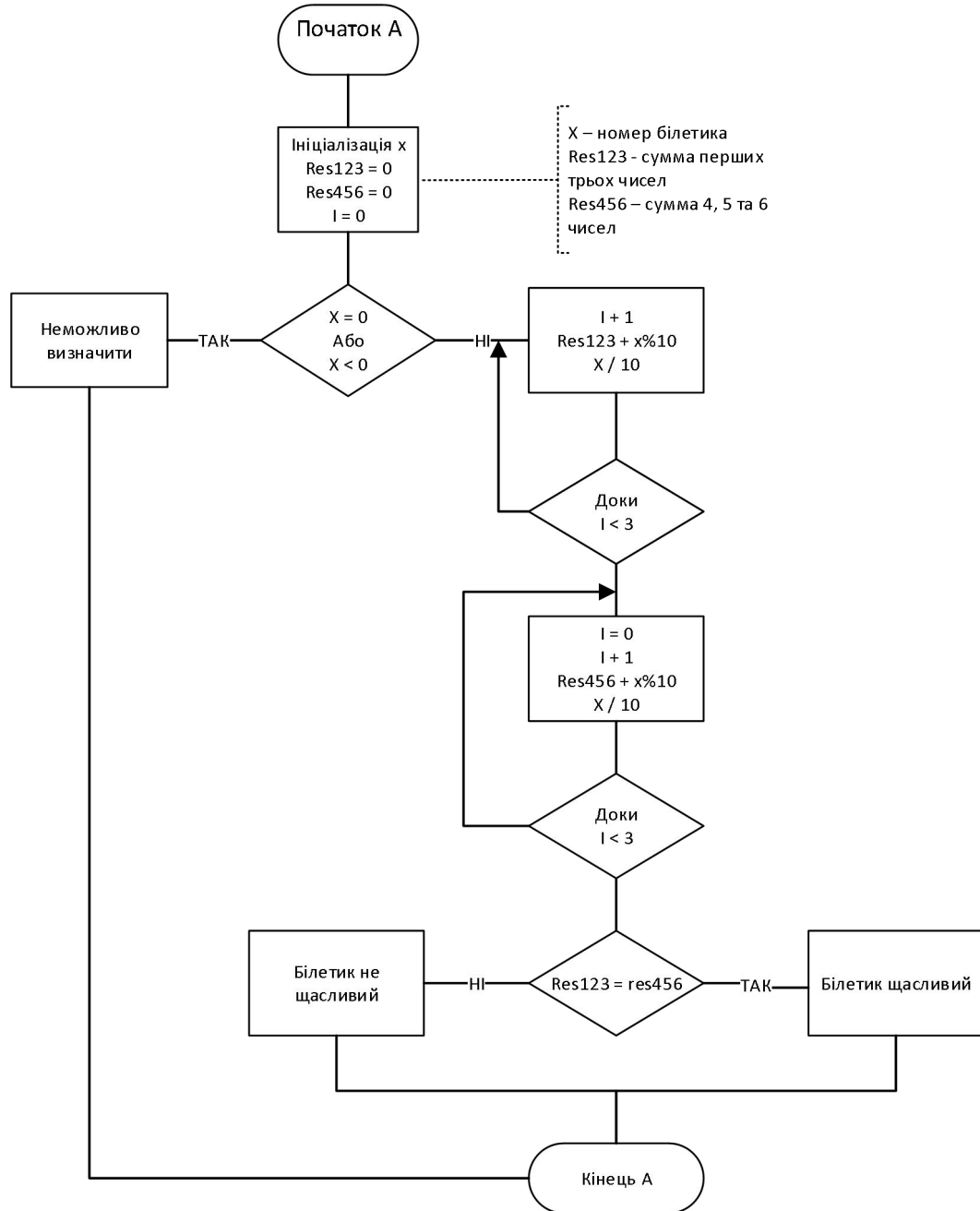


Рисунок 4 - креслення алгоритму

3. Функція визначення простого числа (див рис. 5)

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

Рисунок 5 - 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;

4. Функція визначення ідеально числа (див рис.6)

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

Рисунок 6 - фрагмент коду

Призначення :

-Визначення, чи є число ідеальним;

Аргументи :

-Число;

Опис роботи :

-знаходження дільників числа;

-якщо сума дільників = самому числу - воно ідеальне;

5. Функція знаходження квадратного кореня (див рис 7)

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

Рисунок 7 - 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 8):

$10\sqrt{2} \sim 14.142136$

Рисунок 8 - квадратний корінь з числа 200

6. Функція центрування слова у рядку (див рис 9)

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

Рисунок 9 - 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 10)

__Ivan Bogov__

Рисунок 10 - результат роботи функції 6

Схема алгоритму (див рис 11, 12, 13):

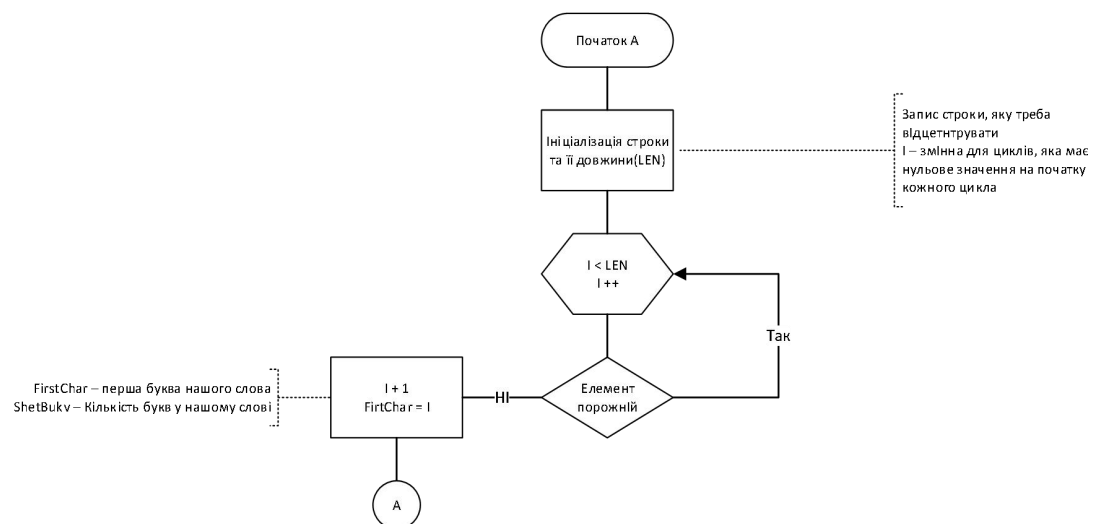


Рисунок 11 - 1 частина креслення алгоритму

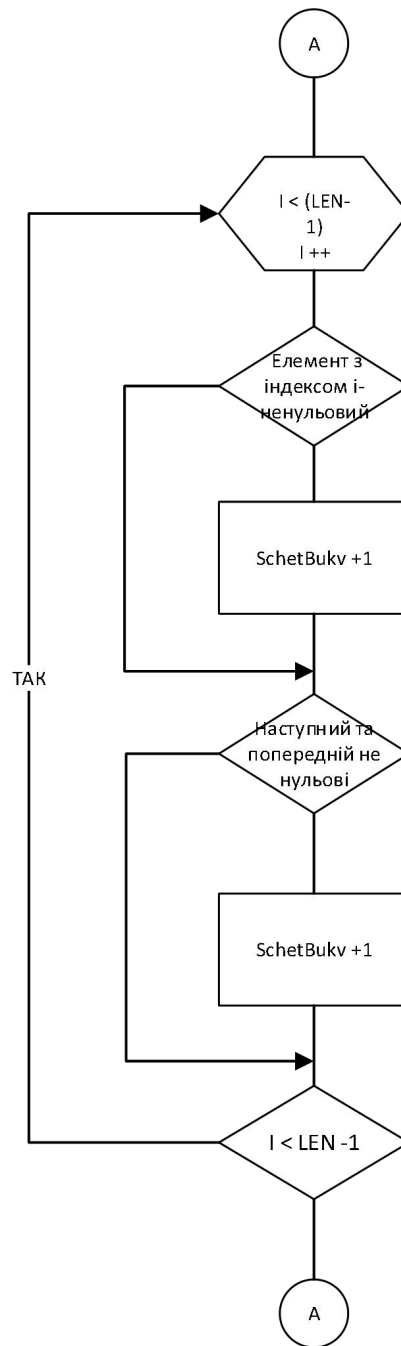


Рисунок 12 - 2 креслення алгоритму

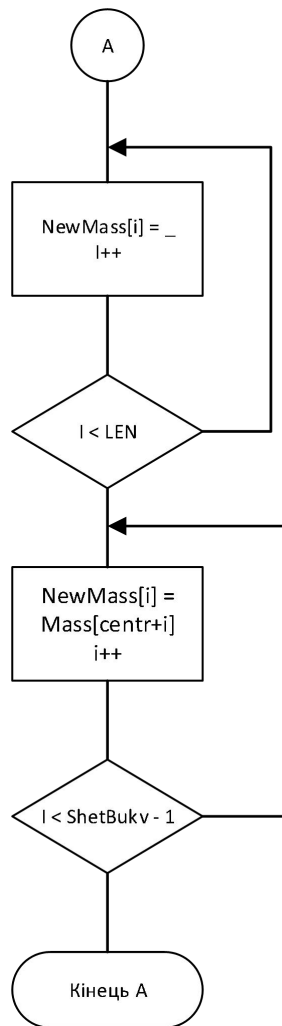


Рисунок 13 - 3 частина креслення алгоритму

7 Функція з варіативною кількістю аргументів (див рис 14)

```

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

```

Рисунок 14 - 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;

Схема алгоритму (див рис 15):

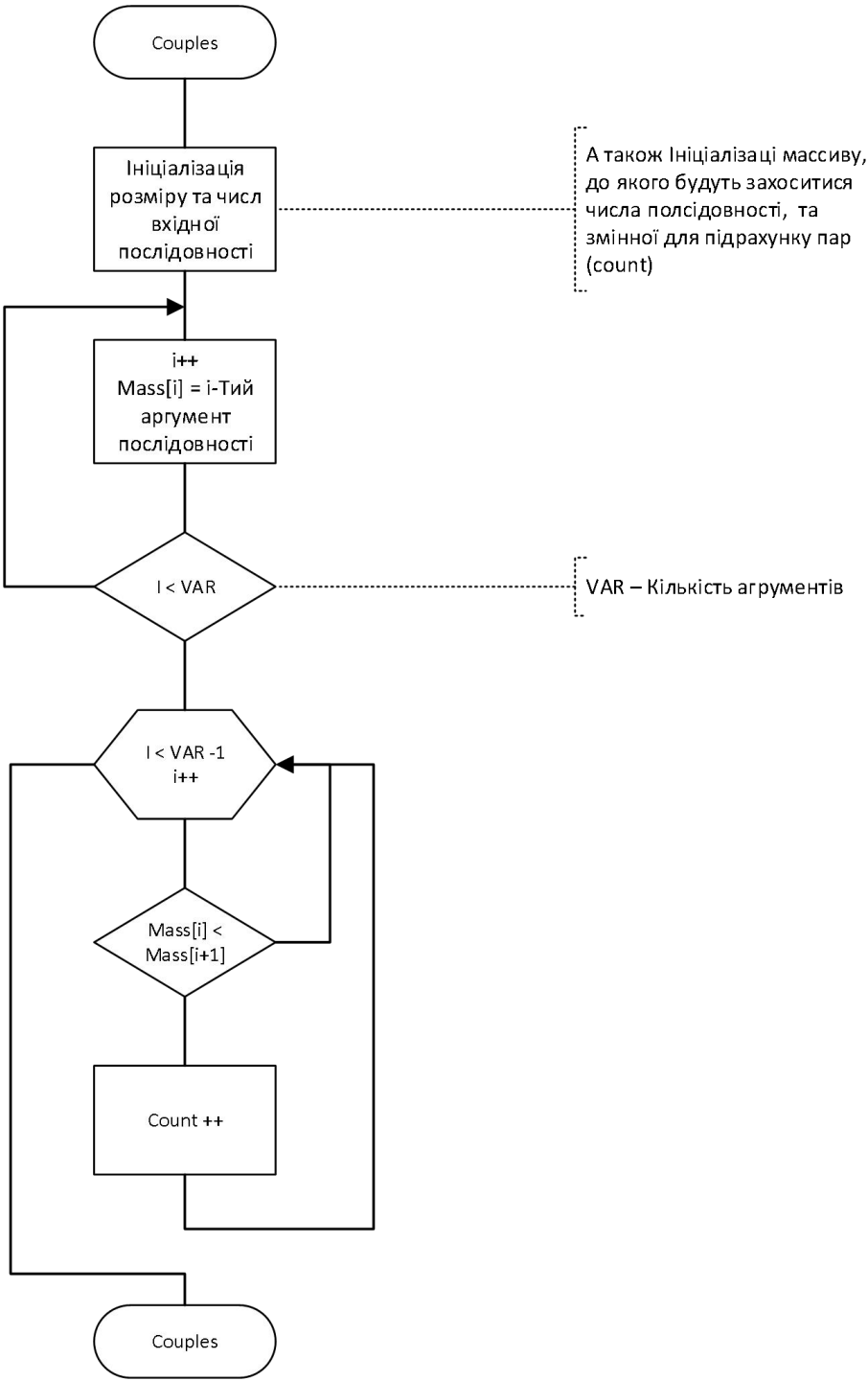


Рисунок 15 - креслення алгоритму

3 Варіанти використання

Для обчислення результатів ми використовуємо gdb - дебагер, влаштований в утиліту gcc.

Щоб подивитися на результат, ми повинні:

- при компіляції вказати рівень інформації для відлатки за допомогою -g;
- відкомпілювати наш файл з кодом;
- запустити його бінарний файл у відлагоднику;
- вказати ім'я функції, як точку зупину щоб подивиться на значення змінних;
- запускаємо програму та вводимо: "info locals";

Висновок:

Протягом даної лабораторної роботи ми навчилися писати повноцінні звіти у форматах doc та markdown.

Переглянути звіт у форматі Markdown можна у папці doc даної лабораторної , яка знаходиться на зовнішньому репозиторії gitlab.