**Лабораторна робота №10**

**Тема:** Вступ до документації проекту

**Розробник**: студентка Зеленець Олена, група КІТ-120а.

**Перевірив:** Челак Віктор Володимирович

**Загальне завдання:** реалізувати програми з використанням функцій.

**Індивідуальне завдання:**

1. Визначити чи є 6-значне число «щасливим квитком»(«щасливий квиток» - це квиток, у якому сума першої половини чисел номера дорівнює сумі другої половини).

2. У заданому тексті знайти кількість слів за умови, що між словами може бути будь-яка кількість пропусків.

3. Реалізувати функцію, що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію з варіативною кількістю аргументів.

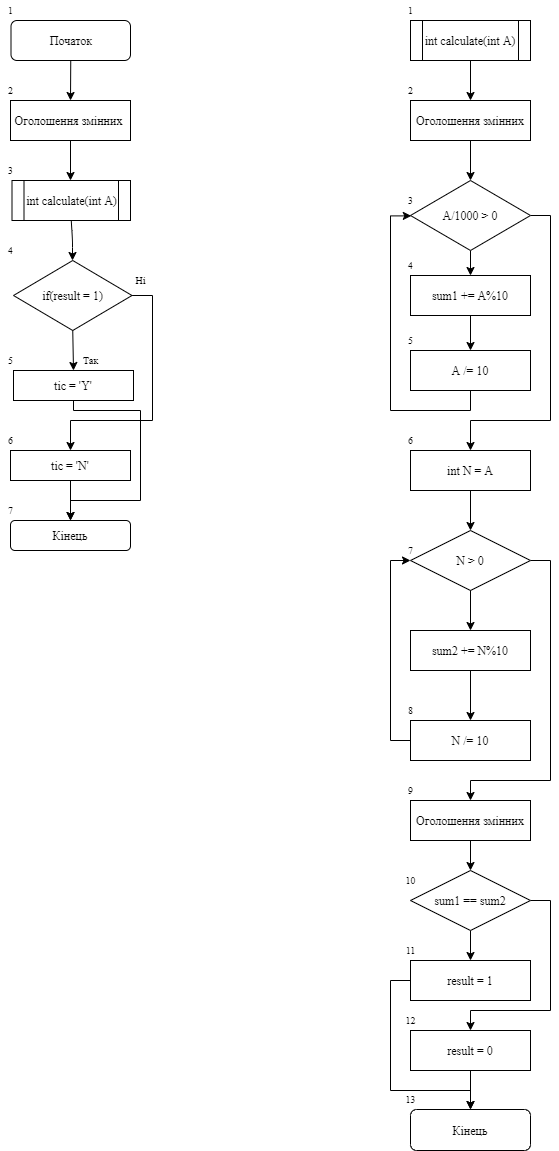
**Опис програми 1:**

* *Функціональне призначення*:

Визначити чи є номер квитка «щасливим», тобто чи дорівнює сума перших та останніх трьох цифр 6-значного числа.

* Опис логічної структури:
* Функція `main` оголошує функцію `calculate`(блок 3) та визначаємо значення змінних та змінну `result`, яка відповідає за визначення результату(блок 2). Далі у змінній `result` викликається наше значення функції і вказуються умови, при яких наш квиток є «щасливим» чи навпаки(блок 4-6). (рис. 1)
* Функція `calculate` виконує алгоритм дій, при яких будуть відокремлюватися перші та останній три цифри нашого 6-значного числа(блок 3-8). І тепер виконується умова, при якій порівнюється sum1 та sum2(блок 10) і в результаті рівності result = 1, тобто true, а коли sum1 не дорівнює sum2, result = 0, тобто false(блок 11-12). (рис. 1)

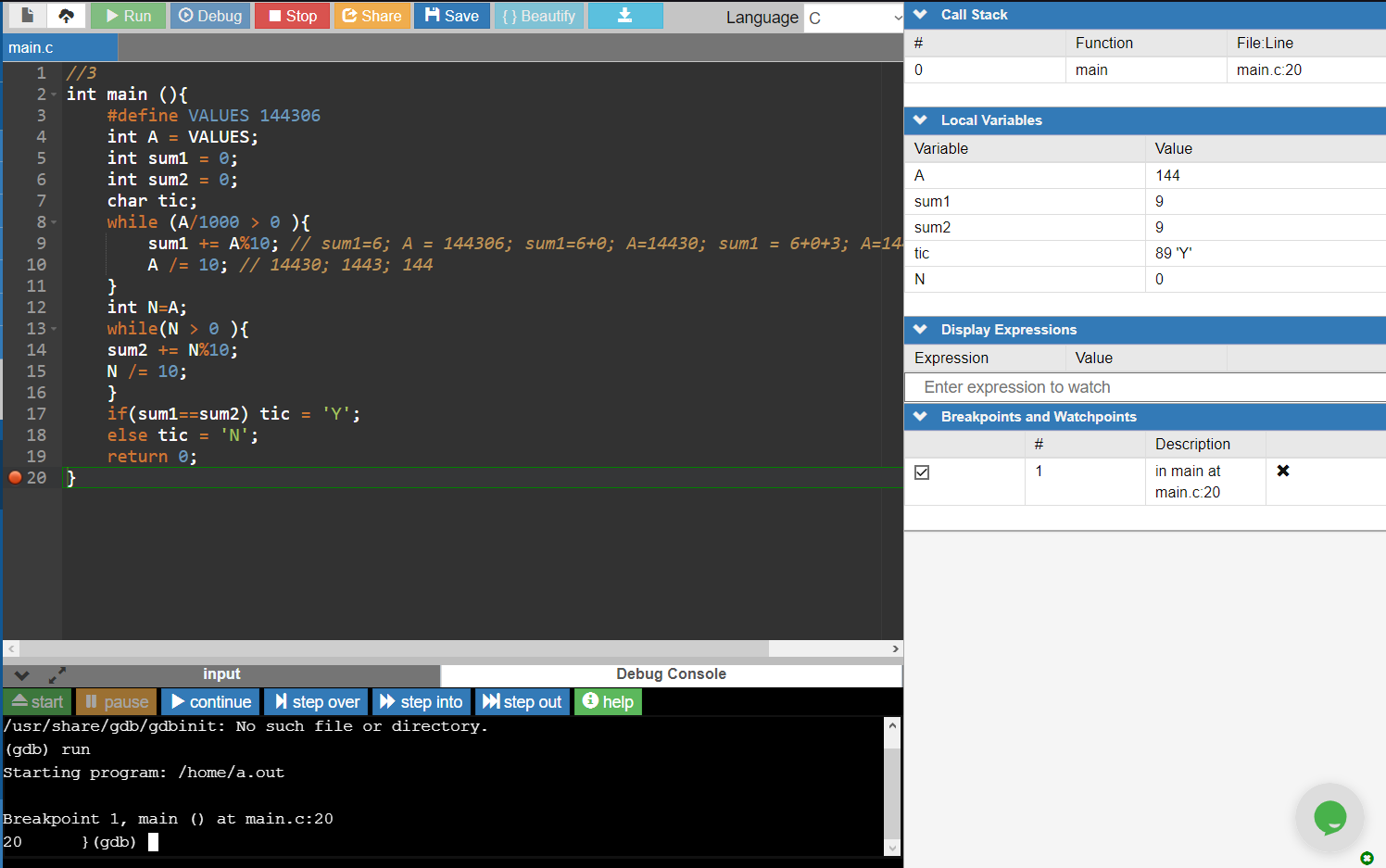
**Схема алгоритму функції `main` та `calculate` наведені на рис. 1.**



**Рисунок 1** – Блок-схема алгоритму програми 1

* *Варіанти використання програми 1:*

Поставимо точку зупинки наприкінці функції та побачимо результат виконання програми(рис. 2).



**Рисунок 2** – Готова програма з постановкою та виведенням результатів

* *Структура програми:*

└── lab10/  
            ├── Doxyfile  
            ├── Makefile  
            ├── README.md  
            ├── doc/  
            │   ├── lab\_10.md  
            │   └── lab\_10.pdf  
            ├──└── src/  
            │       └── main.c

* *Важливі елементи програми:*

Формування змінних. Відокремлення перших та останніх 3 цифр 6-значного числа, порівняння їх сум.

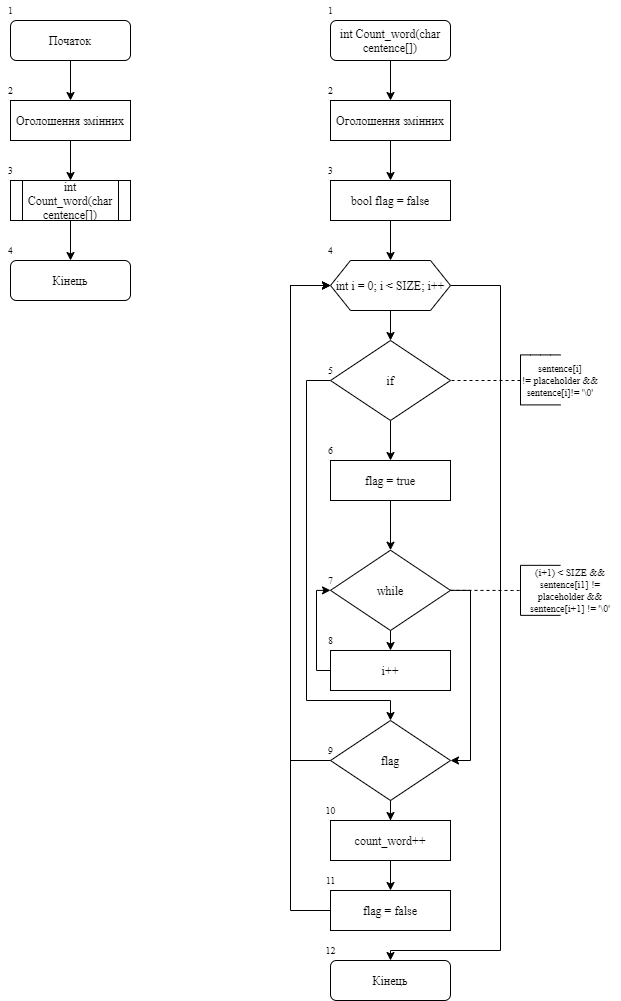
**Опис програми 2:**

* *Функціональне призначення:*

Порахувати кількість слів за умови, що між словами може бути будь-яка кількість пропусків.

* *Опис логічної структури:*
* У функції `main` оголошуємо змінні(блок 2) та функцію `Сount\_word`(блок 3). (рис. 2)
* У функції `Сount\_word` оголошуємо змінні(блок 2), вказуємо значення флагу(блок 3). Описуємо умови, при яких у заданому тексті знаходимо кількість слів(блок 4-9) та в змінну `count\_word` записується результат виконання функції.(блок 10). (рис. 3)

**Схеми алгоритмів функції `main` та `Сount\_word` наведено на рис. 4.**



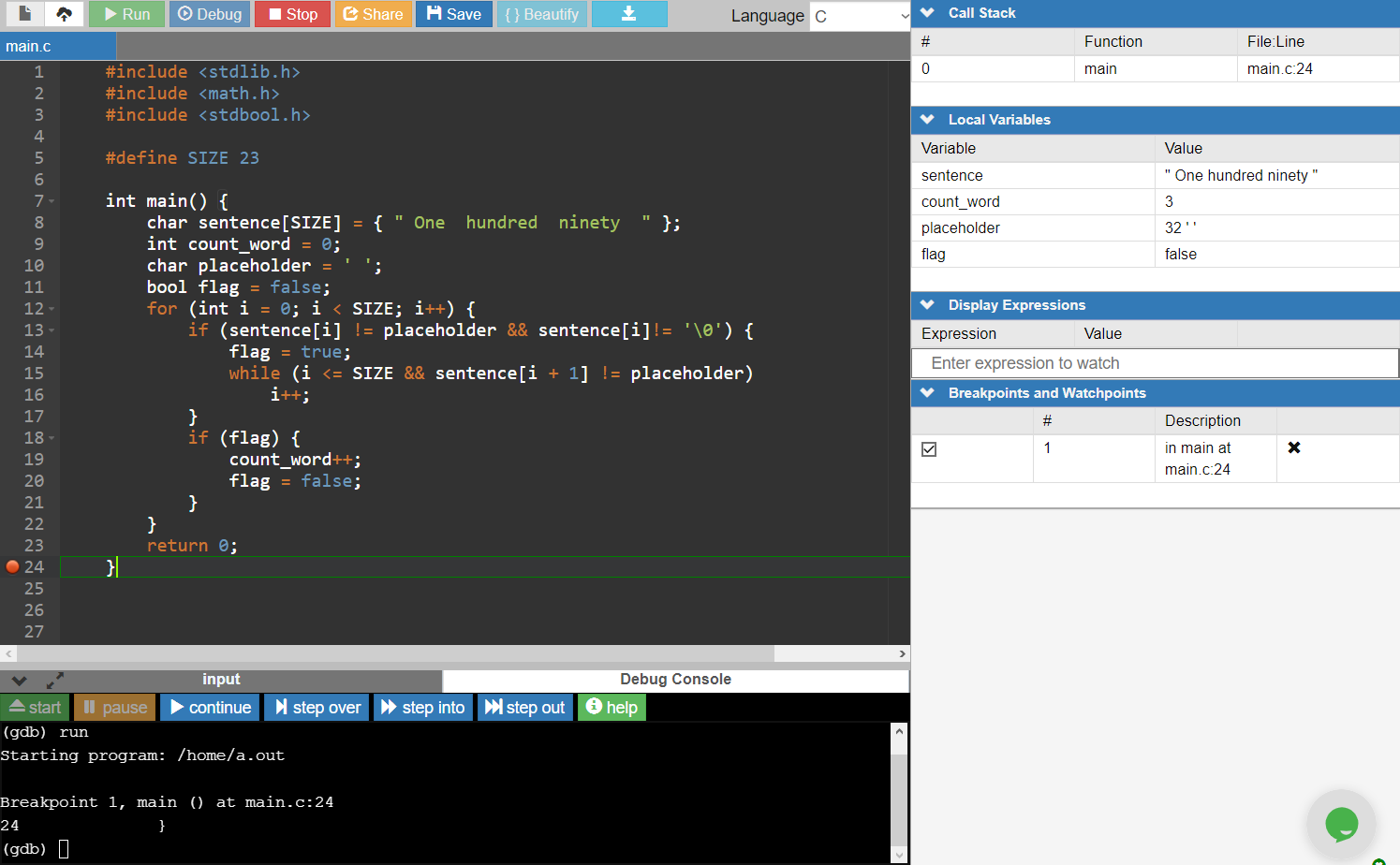
**Рисунок 3** – Блок-схема алгоритму програми 2.

* *Структура програми 2:*

        └── lab10/  
            ├── Doxyfile  
            ├── Makefile  
            ├── README.md  
            ├── doc/  
            │   ├── lab\_10.md  
            │   └── lab\_10.pdf  
            ├──└── src/  
            │       └── main.c

* *Варіанти використання програми 2:*

Поставимо точку зупинки наприкінці функції та побачимо результат виконання програми. (рис.4)



**Рисунок 4** – Готова програма з постановкою та виведенням результатів

* *Важливі елементи програми:*

Оголошення функції та за допомогою неї визначення кількості слів, яке записується у змінну `count\_word`.

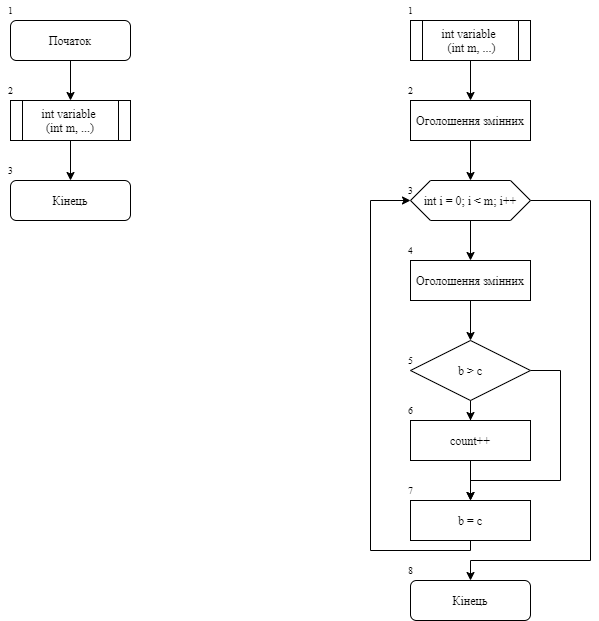
**Опис програми 3:**

* *Функціональне призначення:*

Реалізування функції, що буде визначати кількість пар, у яких перше число менше за наступне.

* *Опис логічної структури:*
* Функція `main` оголошує змінну, що відповідає за функціональне призначення та показує кількість пар, у яких перше число менше за наступне. Викликає функцію `variable`(блок 2). (рис. 3)
* У функцію `variable` оголошуємо змінну, що відповідає за наше функціональне призначення. Далі умови, при яких порівнюється значення з попереднім значенням(блок 3-7). `result` змінюється після перевірки на те, чи задовільна умова.

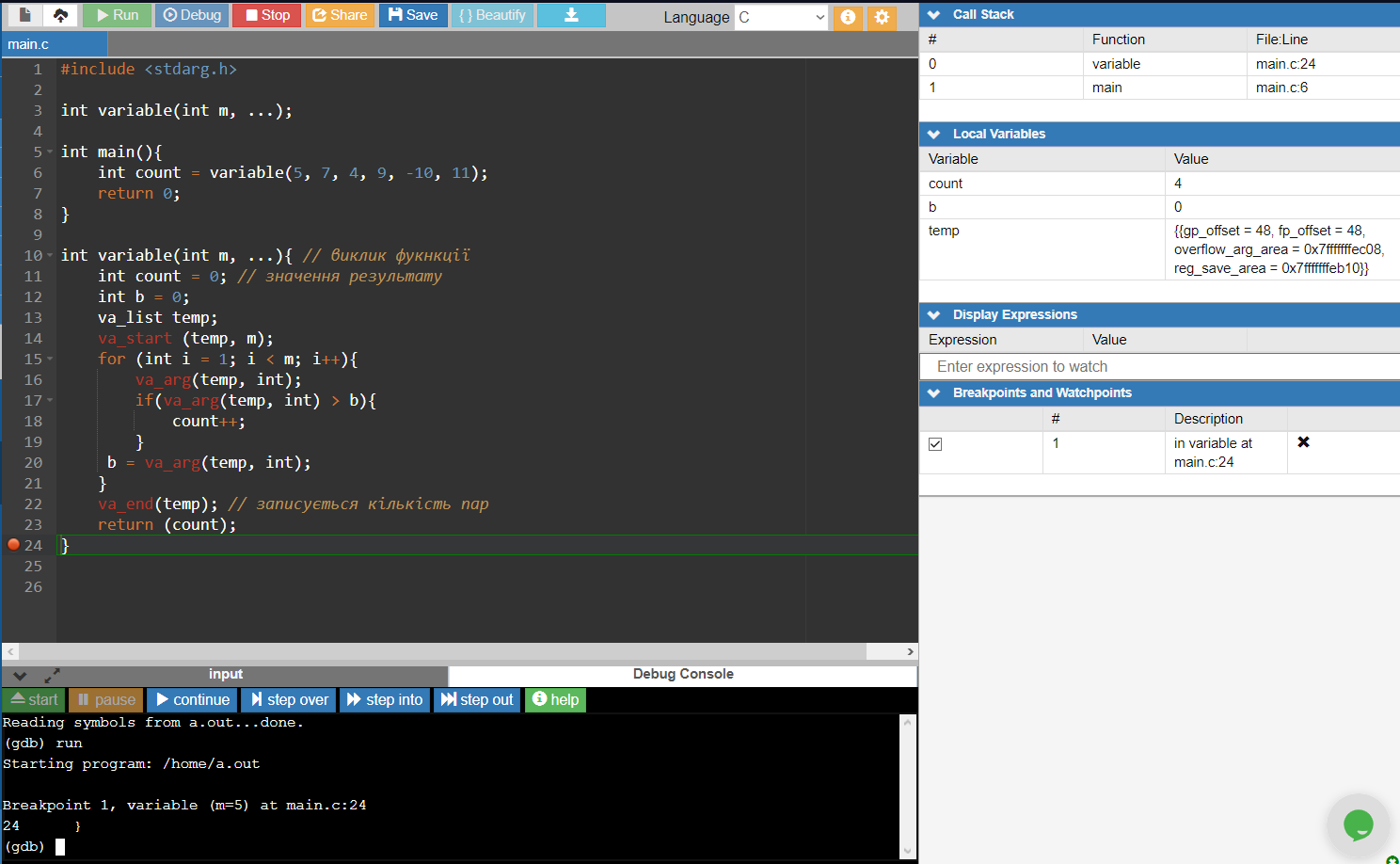
**Схема алгоритмів функції `main` та `variable`наведені на рисунку 5.**



**Рисунок 5** – Блок-схема алгоритму програми 3

* *Структура програми:*  
          └── lab10/  
              ├── Doxyfile  
              ├── Makefile  
              ├── README.md  
              ├── doc/  
              │   ├── lab\_10.md  
              │   └── lab\_10.pdf  
              ├──└── src/  
              │       └── main.c
* *Варіанти використання програми 3:*

Поставивши точку зупинки наприкінці головної функції побачимо результат змінної, яка показує скільки пар чисел, які відповідають заданій умові. (рис. 6)



**Рисунок 6** – Готова програма з постановкою та виведенням результатів

* *Важливі елементи програми:*

Виконання порівняння значення з попереднім значенням.

**Висновок:**

Для виконання лабораторної роботи ми навчились створювати та реалізовувати алгоритми функцій, створювати блок-схеми алгоритмів та оформляти документацію.