

## Лабораторна робота № 5. Циклічні конструкції

### 1. Вимоги

**1.1 Розробник:** Студент групи КІТ-120а Лисенко Микола Валерійович;

**1.2 Перевірив:** Асистент Челак Віктор Володимирович;

**1.3 Загальне завдання:** Виконати усі завдання з категорії "на відмінно".

**1.4 Індивідуальне завдання:** Визначити найбільший спільний дільник для двох заданих чисел.

### 2. Опис програми

**2.1** Було організовано структуру проекту (Рис. 1): створено папку *lab05*, файл *Makefile*, та папку *src* для вихідного коду.

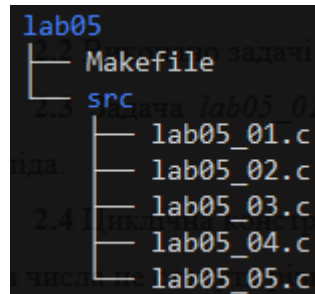


Рисунок 1 – Структура проекту.

**2.2** Виконано задачі 1-5, код збережено у файлах *lab05\_01.c* - *lab05\_05.c*.

**2.3** Задача *lab05\_01.c*. Для визначення НСД було застосовано алгоритм Евкліда.

**2.4** Циклічна конструкція *while* була використана для виконання алгоритму поки числа не стануть рівними (Рис. 2).

```
int main() {  
    int number_a = 360;  
    int number_b = 250;  
    while (number_a != number_b) {  
        if (number_a > number_b) {  
            number_a -= number_b;  
        } else {  
            number_b -= number_a;  
        }  
    }  
    return 0;  
}
```

Рисунок 2 – Реалізація програми.

**2.6** Внесено зміни до *Makefile* для компіляції проекту (Рис. 3).

```

CC = gcc
C_OPTS = -g -std=gnu11 -Wall -Wextra -Wformat

all: clean prep p1 p2 p3 p4 p5

prep:
clean:
    rm -rf dist
prep:
    mkdir dist
p1: lab05_01.bin
p2: lab05_02.bin
p3: lab05_03.bin
p4: lab05_04.bin
p5: lab05_05.bin

lab05_01.bin: src/lab05_01.c
    $(CC) $(C_OPTS) $< -o ./dist/$@
lab05_02.bin: src/lab05_02.c
    $(CC) $(C_OPTS) $< -o ./dist/$@
lab05_03.bin: src/lab05_03.c
    $(CC) $(C_OPTS) $< -o ./dist/$@
lab05_04.bin: src/lab05_04.c
    $(CC) $(C_OPTS) $< -o ./dist/$@
lab05_05.bin: src/lab05_05.c
    $(CC) $(C_OPTS) $< -o ./dist/$@

```

Рисунок 3 – текст Makefile.

2.7 Проект було компільовано, використовуючи *Makefile* та відкрито за допомогою *nemiver* (Рис. 4). Програма виконалася успішно.

```

int main() {
    int number_a = 360;
    int number_b = 250;
    while (number_a != number_b) {
        if (number_a > number_b) {
            number_a -= number_b;
        } else {
            number_b -= number_a;
        }
    }
    return 0;
}

```

Variable	Value	Type
▼ Local Variables		
number_a	10	int
number_b	10	int
Function Arguments		

Рисунок 4 – Відлагодження програми з *nemiver*.

**2.8** Зміни було зафіксовано у *git*-репозиторії *Programming*.

### **3. Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду використання циклічних конструкцій, роботи з *Makefile* та системою контролю версій *git*.