

Міністерство освіти і науки України  
Національний технічний університет  
«Харківський політехнічний інститут»  
Кафедра «Обчислювальна техніка та програмування»

### **ЗВІТ**

Про виконання лабораторної роботи № 5  
«Циклічні конструкції»

Керівник: викладач  
Бульба С. С.

Виконавець: студент гр. КІТ-120в  
Бабенко А. П.

Харків 2020

# Лабораторна робота №5. Циклічні конструкції

## 1. Вимоги

### 1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 29 листопада 2020;

### 1.2 Загальне завдання

**На оцінку “відмінно”.** Необхідно виконати усі завдання з даної категорії (проте звіт та відповідні зміни до системи контролю версіями виконуються лише за одним обраним студентом варіантом).

1. Визначити найбільший спільний дільник для двох заданих чисел.
2. Визначити, чи є задане ціле число простим.
3. Визначити, чи є ціле 6-значне число «щасливим» квитком («щасливий квиток» – квиток, в якому сума першої половини чисел номера дорівнює сумі другої половини. Наприклад, білет з номером *102300* є щасливим, бо  $1 + 0 + 2 = 3 + 0 + 0$ ).
4. Визначити, чи є задане число досконалим (якщо воно дорівнює сумі своїх дільників). Наприклад, 6 - досконале число, бо  $6 = 1 + 2 + 3$ .
5. Без допомоги зовнішніх бібліотек, отримати корінь заданого числа.

### 1.3 Індивідуальне завдання

Визначити, чи є задане число простим.

## 2 Виконання роботи

2.1 Створення файлу з вихідним кодом. Написання коду і коментарів до нього. Зображено на рис.1.

```

int isSimple(int);

int main()
{
    printf("%d\n", isSimple(17));
    return 0;
}

int isSimple(int n) {
    int res = 0;

    for (int i = 2; i <= n; i++)
    {
        if (n % i == 0)
        {
            res += 1;
        }
    }

    if (res > 1)
    {
        res = 0;
    }
    return res;
}

```

Рисунок 1 — код програми

2.2 Компіляція проекту за допомогою команди “make clean prep compile”. Зображено на рис.2.

```

anton@anton-X55VD:~/dev/Programming-repo/lab05$ make clean prep compile
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/task1.c -o ./dist/task1.bin

```

Рисунок 2 — компіляція проекту

2.3 Відкрито у відладнику nemiver виконуючий файл main.bin. Ставимо точку зупину, проходимо по файлу і бачимо зміну значення змінної в момент проходження по циклу. Зображено на рис.3.

Переменная	Значение	Тип
▼ Локальные переменные		
res	1	int
i	17	int
▼ Параметры функции		
n	17	int

Рисунок 3 — файл у відлагоднику

2.4 Створення блок-схеми програми. Зображення блок-схеми на рис.4.

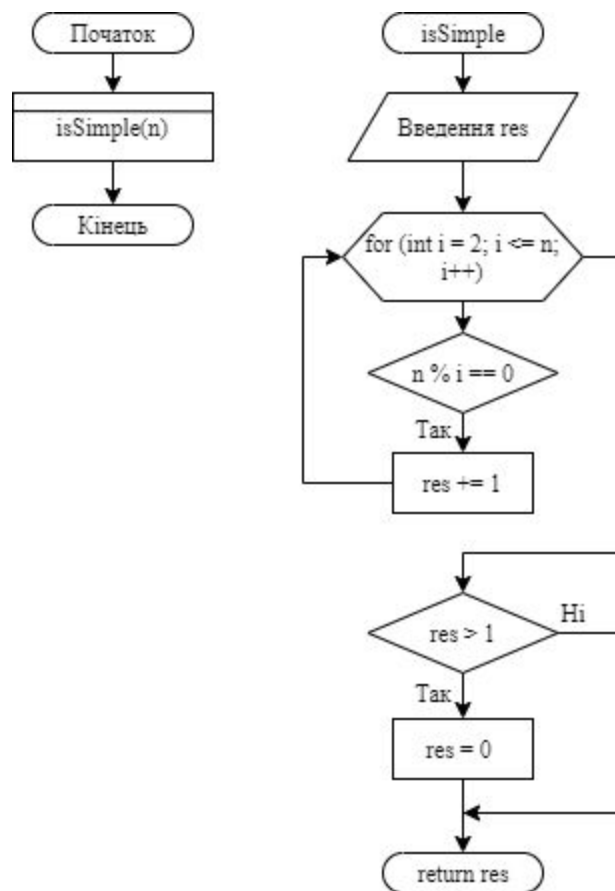


Рисунок 4 — блок-схема

2.10 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 5, 6.

## Опис функцій

---

### ◆ isSimple()

```
int isSimple ( int n )
```

Перевірка на простоту

Функція, що визначає чи є задане число простим

#### Аргументи

**n** число яке буде перевірятись на простоту

#### Повертає

Результат роботи функції (res)

### ◆ main()

```
int main ( )
```

Головна функція

Послідовність дій:

- виклик функції isSimple

#### Повертає

успішний код виконання програми (0)

Граф всіх викликів цієї функції:



Рисунок 5, 6 — опис функцій

## Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм з циклічними конструкціями, а також створено програму, що визначає, чи є задане число простим.

