

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Обчислювальної техніки та програмування»

Звіт з лабораторної роботи №5

Тема: «Циклічні конструкції»

Виконав:

ст. гр. КІТ-120в Львов Артем

Перевірив:

Бульба С.С.

Харків 2020

Лабораторна робота №5. Циклічні програми

Розробник:

Львов Артем Сергійович

Студент групи КІТ-120В

16.12.2020

Загальне завдання:

Написання циклічних програм

Обчислення варіанту індивідуального завдання:

За формулою $N_t = ((N_j - 1) \% C) + 1$:

$$N_t = ((12 - 1) \% 5) + 1 = 2;$$

Індивідуальне завдання:

Визначити, чи є задане число простим

Функціональне призначення:

Програма отримує певне число, та обчислює, чи є воно простим (тобто ділиться націло тільки на себе і на одиницю).

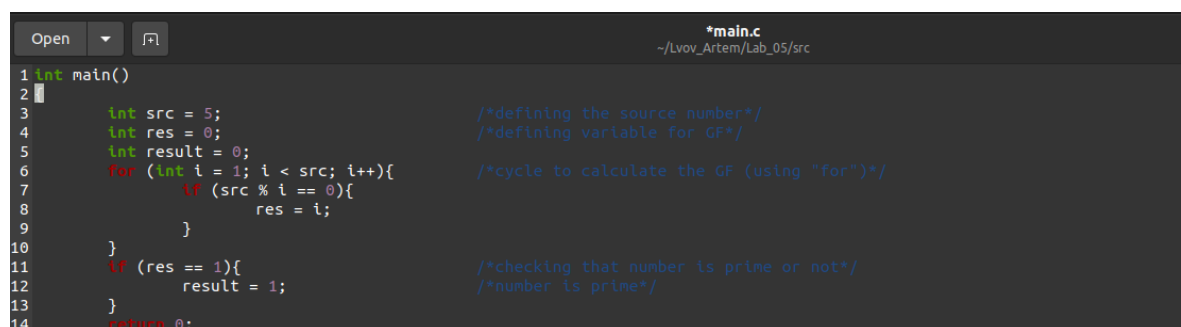
Виконання роботи:

1. Створення репозиторію:

Створюю новий репозиторій Lab_05, імпортую туди репозиторій Давидова В.В та клоную до локального репозиторію командою git clone.

2. Написання коду:

Пишу код програми у файлі Lab_05/src/main.c



```
*main.c
~/Lvov_Artem/Lab_05/src

1 int main()
2
3     int src = 5;                /*defining the source number*/
4     int res = 0;                /*defining variable for GF*/
5     int result = 0;
6     for (int i = 1; i < src; i++){ /*cycle to calculate the GF (using "for")*/
7         if (src % i == 0){
8             res = i;
9         }
10    }
11    if (res == 1){                /*checking that number is prime or not*/
12        result = 1;              /*number is prime*/
13    }
14    return 0;
```

Рисунок 2 – код програми

2.1. Компіляція проекту:

Компілюю проект за допомогою команди `make clean prep compile`:

```
o71@071:~/Lab_05$ 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 src/main.c -o ./dist/main.bin
o71@071:~/Lab_05$
```

Рисунок 2.1 – компіляція проекту

2.2. Перевірка за допомогою nemiver:

Перевіряю роботу проекту командою `nemiver ./dist/main.bin`

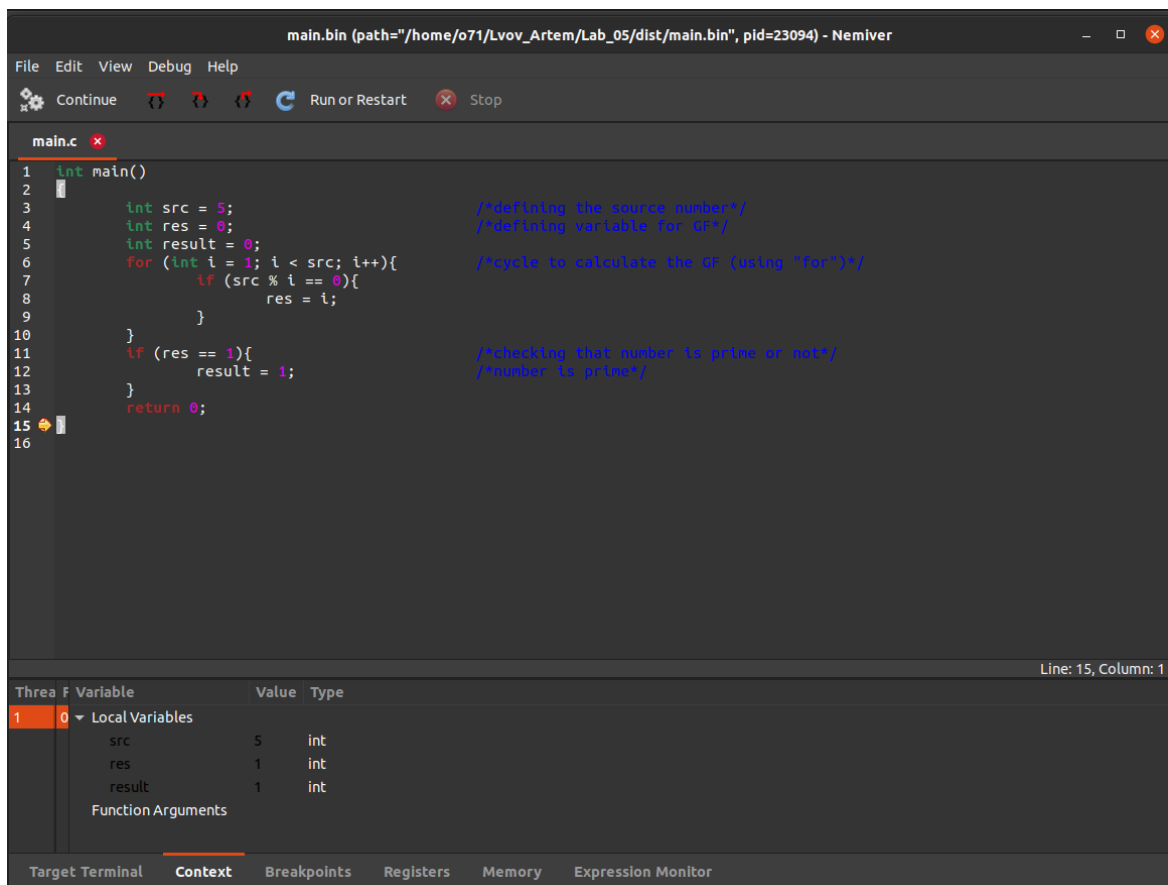


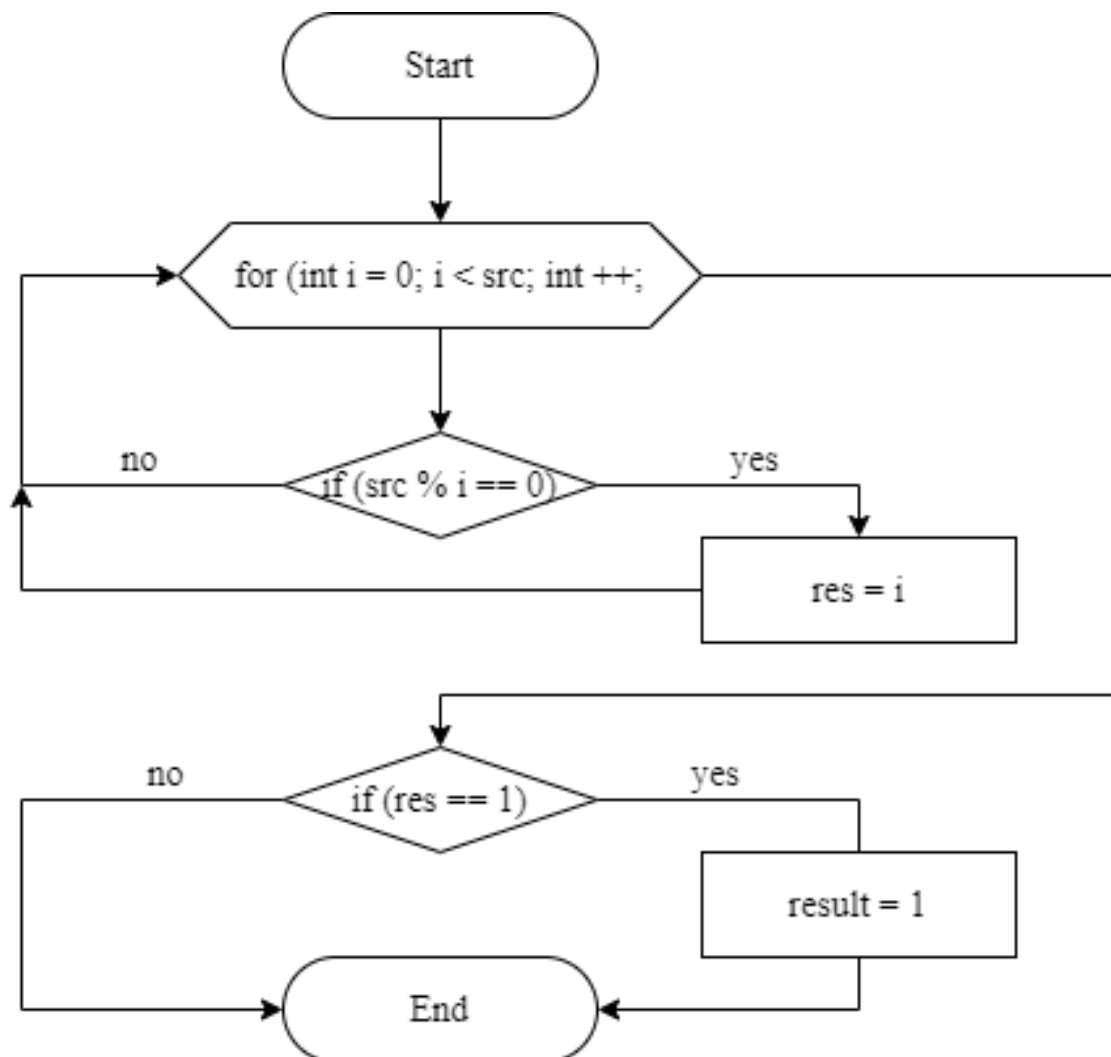
Рисунок 2.2 – вікно nemiver

3. Завантаження проекту на GitHub:

```
^Co71@071:~/Lab_05$ git add .
o71@071:~/Lab_05$ git commit -m"Lab_05"
[master 24249e5] Lab_05
 2 files changed, 26 insertions(+), 15 deletions(-)
 rewrite src/main.c (93%)
 create mode 100644 src/main1.c
o71@071:~/Lab_05$ git push
Username for 'https://github.com': sudo-071
Password for 'https://sudo-071@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 635 bytes | 635.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/sudo-071/Lab_05
 6b6b022..24249e5 master -> master
```

Рисунок 3 – Виконання команд git add, git commit, git push

4. Наведу графічну блок-схему роботи даної програми:



Висновки:

Під час розробки цієї програми я навчився користуватися циклічними конструкціями на прикладі циклу `for` та написав програму що дозволяє визначити, чи є певне число простим.