

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Обчислювальної техніки та програмування»

Звіт з лабораторної роботи №4

Тема: «Розробка програм, що розгалужуються»

Виконав:

ст. гр. КІТ-120в Львов Артем

Перевірив:

Бульба С.С.

Харків 2020

Лабораторна робота №4. Розробка програм, що розгалужуються

Розробник:

Львов Артем Сергійович

Студент групи КІТ-120В

09.12.2020

Загальне завдання:

Розробка програм, що розгалужуються

Обчислення варіанту індивідуального завдання:

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

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

Індивідуальне завдання:

Визначити, у скільки разів значення дробової частини числа більше за цілу. Організувати перевірку ділення на 0. Результат “обрізати” до другого знаку після коми.

Функціональне призначення:

Програмі задається певне число з дробовою частиною. Задача програми - визначити окремо дробову і цілну частину числа, поділити дробову на цілу та вивести в змінній result значення результату ділення, округлене до двох знаків після коми. Звісно, включаючи перевірку від ділення на 0.

Виконання роботи:

1. Створення репозиторію:

Створюю новий репозиторій Lab_04, імпортую туди репозиторій Давидова В.В та клоную до локального репозиторію командою git clone.

2.Написання коду:

Пишу код програми у файлі Lab_04/src/main.c



```
1 int main()
2 {
3     double src = 123.656;           /*defining the source number*/
4     int int_part = src;              /*extracting integer part of the number*/
5     int frac_part = (src - int_part) * 1000; /*extracting fractional part of the number*/
6     double result;
7     if (int_part != 0){               /*dividing by zero check*/
8         double res = ((float)frac_part / (float)int_part);
9         int res_2 = res * 100;
10        result = res_2 / 100.0;
11    }else{
12        result = 0; /*couldn't divide by zero*/
13    }
14    return 0;
15 }
```

Рисунок 2 – код програми

2.1. Компіляція проекту:

Компілюю проект за допомогою команди make clean prep compile:



```
o71@071: ~$ cd Lab_04
o71@071:~/Lab_04$ 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/lib.c src/main.c -o ./dist/main.bin
src/main.c: In function 'main':
src/main.c:4:17: warning: conversion from 'double' to 'int' may change value [-Wfloat-conversion]
  4 | int int_part = src; /*extracting integer part of the number*/
    |               ^
src/main.c:5:18: warning: conversion from 'double' to 'int' may change value [-Wfloat-conversion]
  5 | int frac_part = (src - int_part) * 1000; /*extracting fractional part of the number*/
    |               ^
src/main.c:9:15: warning: conversion from 'double' to 'int' may change value [-Wfloat-conversion]
  9 | int res_2 = res * 100;
    |             ^
src/main.c:6:9: warning: variable 'result' set but not used [-Wunused-but-set-variable]
  6 | double result;
    |         ^
gcc -std=gnu11 -g -Wall -Wextra -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src/lib.c test/test.c -o ./dist/test.bin
o71@071:~/Lab_04$
```

Рисунок 2.1 – компіляція проекту

2.2. Перевірка за допомогою nemiver:

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

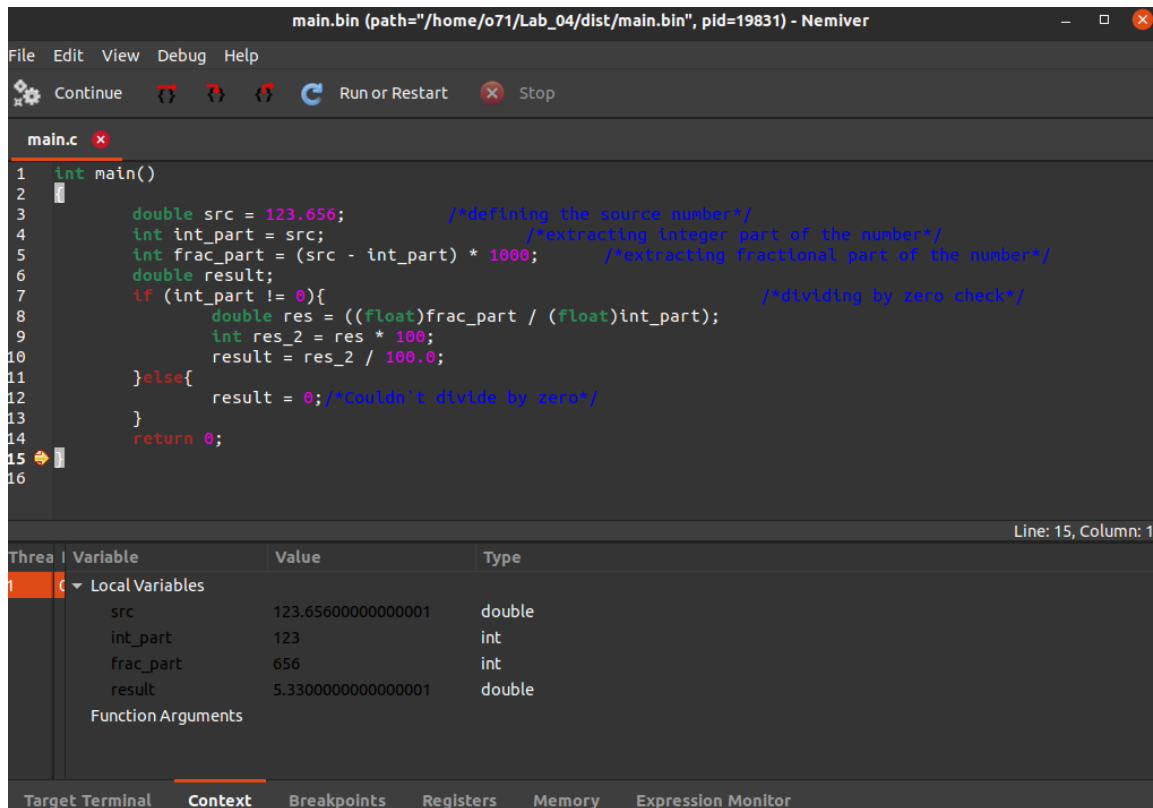


Рисунок 2.2 – вікно nemiver

3. Завантаження проекту на GitHub:

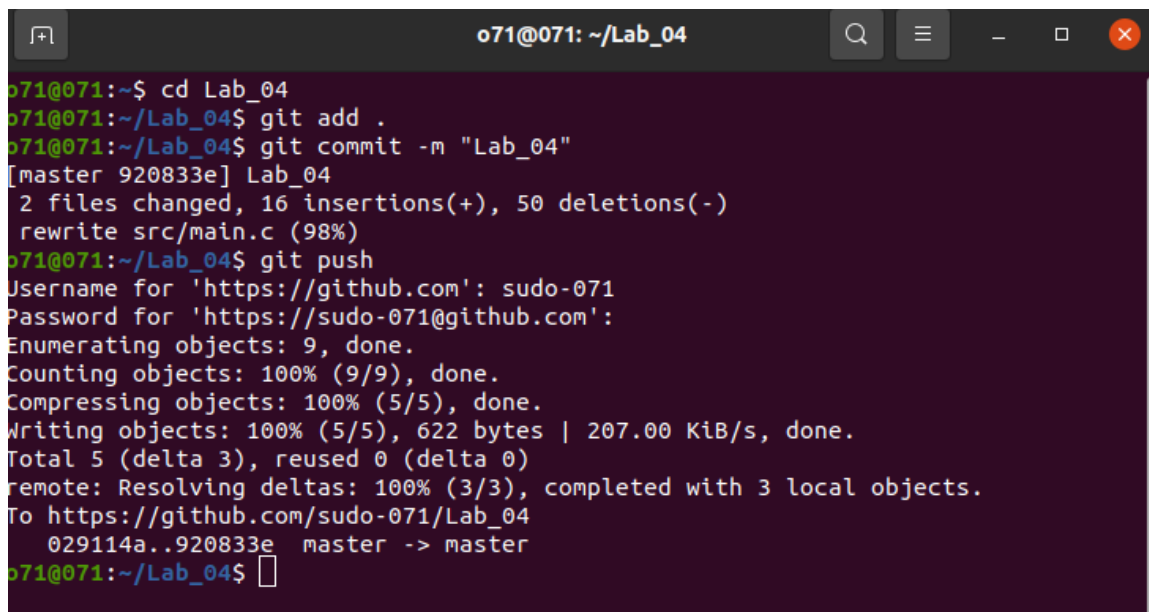
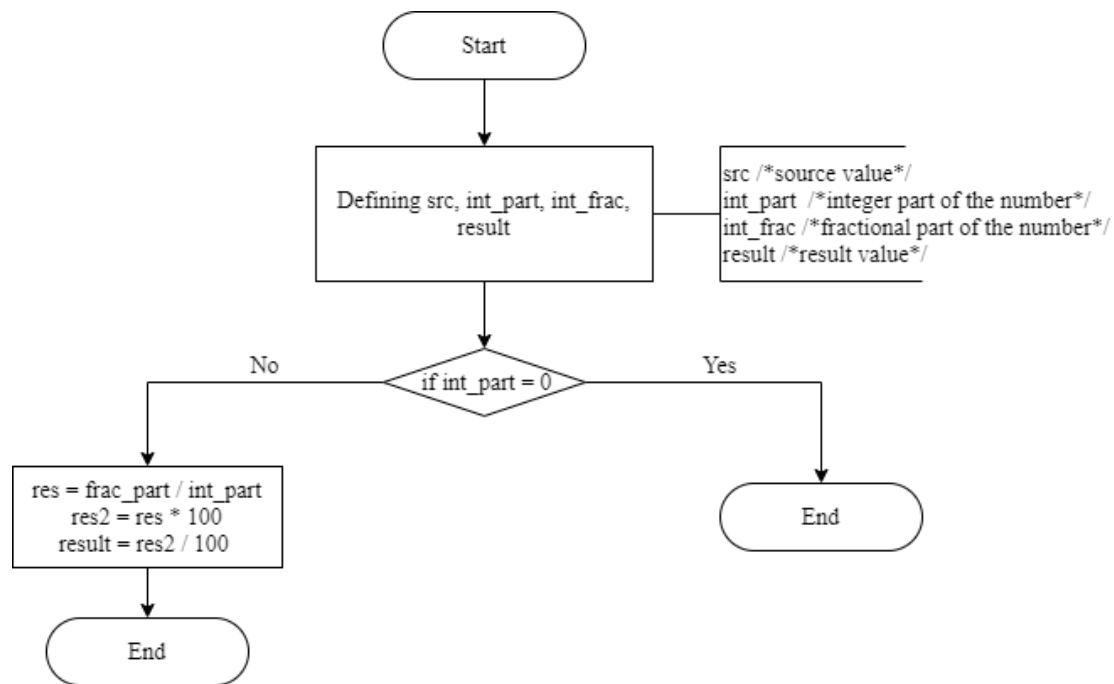


Рисунок 3 – Виконання команд git add, git commit, git push

4. Наведу графічну блок-схему роботи даної програми:



Висновки:

Розробляючи цю програму, я вивчив основні принципи роботи програм, що розгалужуються на прикладі перевірки на ділення на 0. Написав програму, що дозволяє легко визначити, в скільки разів дрібна частина більша за цілу в певному числі. Якщо його ціла частина дорівнює нулю, звісно.