**Лабораторна робота № 3 . Розробка лінійних програм**

**1 ВИМОГИ**

* 1. **Розробник**

-Курінна Єлизавета Ігорівна;

-студентка групи КІТ – 121г;

-16-жов-2021.

**1.2 Загальне завдання**

Розробити лінійну програму

**1.3 Індивідуальне завдання**

За заданим опором трьох резисторів r1, r2, r3, які з’єднані паралельно, визначити загальний опір.

1. **ОПИС РОБОТИ**
   1. **Робота з Makefile та README.md**

У методичці було написано, що можна користуватись наданим файлом Makefile з першої лабораторної. Тому копіюю його в свою папку за допомогою команди cp:

***mkdir lab03***

***cp sample\_project/lab00/Makefile lab03***

Проте слід виконати зміни в самому файлі, для цього я виконала наступний алгоритм дій та закоментувала 2 рядок( де декларується змінна LAB\_OPTS):

***cd lab03***

***vim Makefile (/\* LAB\_OPTS = -I./src src/lib.c \*/)***

Наступною дією було закомінити зміни:

***git status***

***git add Makefile***

***git commit –m “Deleted file usage”***

***git push***

Робота з файлом README.md

***vim README.md ( де вписала номер лабораторної, назву та завдання)***

***git status***

***git add README.md***

***git commit –m “Created file with the laboratory number, name and task"***

***git push***

* 1. **Написаннякоду**

Створюю файл за допомогою команди vim

***mkdir src***

***cd src***

***vim main.c***

Моїм завданням було знайти загальний опір резистора, який знаходиться за формулою: R = 1/r1 + 1/r2 + 1/r3 (r1 = 2.5, r2 = 10, r3 = 40) для цього я прописала такий код:

*/\**

*task:*

*For a given resistance of three resistirs r1, r2, r3, which are connected in parallel, determine total resistance.*

*find total resistance in a parallel circuit: total = 1/r1 + 1/r2 + 1/r3*

*input data:*

*-r1 - the resistance of the first resistor, float*

*-r2 - the resistance of the second resistor, float*

*-r3 - the resistance of the third resistor, float*

*output data :*

*- total resistance, float*

*\*/*

*int main () /\* For a given resistance of three resistirs r1, r2, r3, which are connected in parallel, determine total resistance.*

*1. Enter variables and assign them a data type*

*2. Use the formula of the total resistance: total = 1/r1 + 1/r2 + 1/r3*

*3. Rewrite the formula: total = R1 + R2 + R3, where R1 = 1 / r1, R2 = 1 / r2, R3 = 1 / r3 \*/*

*{*

*float r1 = 2.5, r2 = 10, r3 = 40, R1, R2, R3, total;*

*R1 = 1 /(float) r1; /\* 1 / 2.5 = 0.4 \*/*

*R2 = 1 /(float) r2; /\* 1 / 10 = 0.1 \*/*

*R3 = 1 /(float) r3; /\* 1 / 40 = 0.025 \*/*

*total = R1 + R2 + R3; /\* find total resistance in a parallel circuit 0.4 + 0.1 + 0.025 = 0.525 \*/*

*return 0;*

*}*

Якщо виконати обчислення самостійно, то результат повинен бути 0.525.

Виконаємо компіляцію за допомогою команди:

***clang –g main.c –o main.bin***

***Тепер провіримо програму через відлагодник:***

***lldb main.bin***

***l 1***

***l***

***l***

***b 30***

***run***

***print total***

для підтвердження коректності роботи програми, зупинила відлагодник на строці з "return 0" та введемо команду "p total". Після вводу команди я отримала:

(lldb) p total

(float) $0 = 0.524999976

З цього етапу випливає, що результат майже співпав, адже похибка 0.000000024 є дозволеною при виконанні операцій з плаваючою крапкою. Подібність результатів говорить про те, що програма працює коректно.

Наступним кроком я закомітила це:

***git status***

***git add lab03/src/main.c***

***git commit –m “Created a code with comments”***

***git push***

* 1. **Точка входу**

Точка входу починається з функції main(), з якої й починається виконання програми. Якщо точку входу не знайдено або вона не єдина, то отримаємо помилку на етапі компіляції

* 1. **Структура директорії**

**tree**

├── doc

│   └── lab03.txt

├── Makefile

├── README.md

└── src

└── main.c

Висновки

Завдяки цій лабораторній роботі, я навчилась закоментовувати рядки, писати код для лінійних програм, перевіряти чи працює програма коректно за допомогою відлагодника, дізналась що таке точка входу, де вона знаходиться, її призначення та чому вона лише одна.