

Лабораторна робота №4. Розробка програм, що розгалужуються

Індивідуальне завдання на оцінку “відмінно” номер 3.

1. Вимоги

1) Розробник

- Князькін Владислав Ігорович
- студент групи КІТ-320
- 31.10.20

2) Загальне завдання

Дано три числа k , m , n . Змінити значення змінних таким чином, щоб виконувалась умова $k < m < n$.

2. Хід роботи

1. Зробимо у раніше створеному репозиторії нову під-директорію *lab04*.

```
valadon@valadon-VirtualBox:~/naxxramass $ mkdir lab04
```

2. Створимо папку для нашого завдання *lab04_3* та відкриємо у запусимо у ньому *nano*.

```
valadon@valadon-VirtualBox:~/naxxramass $ mkdir lab04_3
```

```
valadon@valadon-VirtualBox:~/naxxramass/lab04/lab04_3 $ nano
```

```
valadon@valadon-VirtualBox: ~/naxxramass/lab04/lab04_3
```

```
GNU nano 4.8          Новый буфер
```

```
[ Вас приветствует nano. Для получения справки, нажмите Ctrl+G. ]
```

^G Помощь	^O Записать	^W Поиск	^K Вырезать	^J Выводить	^C ТекПозиц
^X Выход	^R ЧитФайл	^_ Замена	^U Paste Text	^T Словарь	^_ К строке

3. Записуємо наш код програми, дотримуючись всіх відступів та закриття дужок.

```
int main () { // * Створюємо функцію main

    int k = 28; // * Створюємо локальну змінну k
    int m = 64; // * Створюємо локальну змінну m
    int n = 15; // * Створюємо локальну змінну n

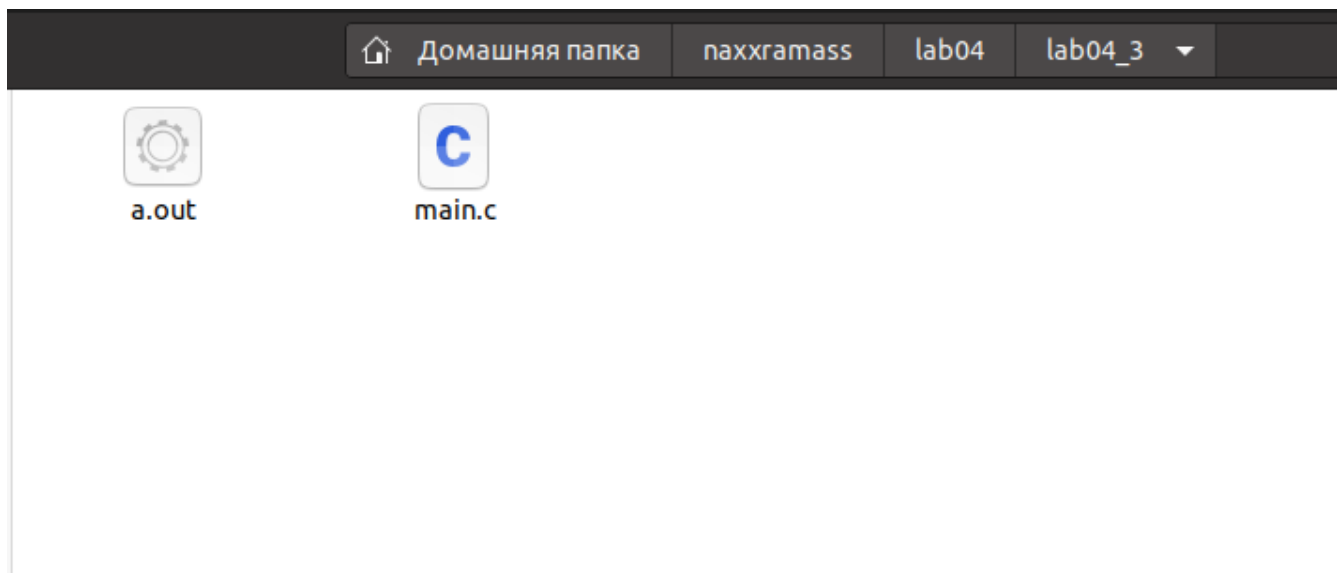
    int temp; // * Створюємо змінну temp для обміну значень, "локальний буфер обміну"
    if (k > m) { // * Створюємо перша умова if, якщо k більше m
        temp = k; // * Порівнюємо temp з k - перший об'єкт для обміну готовий
        k = m; // * Порівнюємо k і m - обмінюємо їх значеннями, отже і місцями
        m = temp; // * Порівнюємо temp з m - другий об'єкт для обміну готовий
    } // * Завершення першого умови
    if (k > n) { // * Створюємо друга умова if, якщо k більше n
        temp = k; // * Порівнюємо temp з k - перший об'єкт для обміну готовий
        k = n; // * Порівнюємо k і n - обмінюємо їх значеннями, отже і місцями
        n = temp; // * Порівнюємо temp з n - другий об'єкт для обміну готовий
    } // * Завершення другого умови
    if (m > n) { // * Створюємо третя умова if, якщо m більше n
        temp = m; // * Порівнюємо temp з m - перший об'єкт для обміну готовий
        m = n; // * Порівнюємо m і n - обмінюємо їх значеннями, отже і місцями
        n = temp; // * Порівнюємо temp з n - другий об'єкт для обміну готовий
    } // * Завершення третього умови
    return 0; // * Закінчуємо поточну функцію main () і повертаємо результат програми
}
```

4. Збережемо наш код та підпишемо його *main.c*.

5. Повертаємось до терміналу. Завдяки команді (у каталозі lab04):

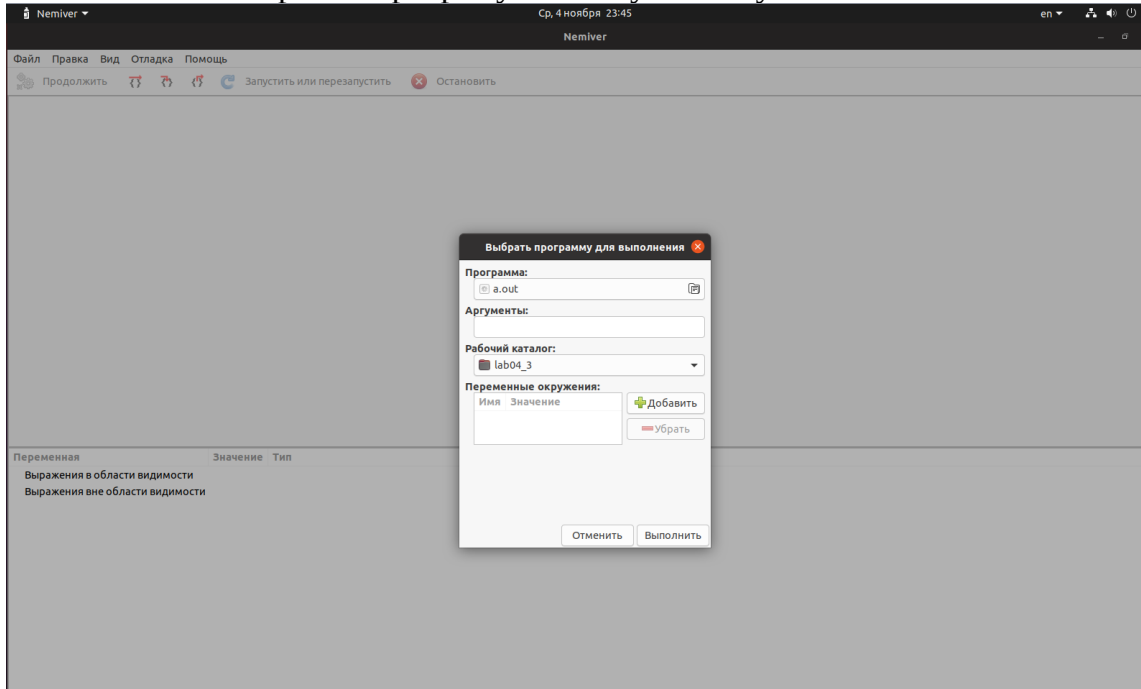
```
gcc -g main.c    /* Зробимо виконувальний файл для нашого
main.c
```

Він автоматично назветься *a.out*



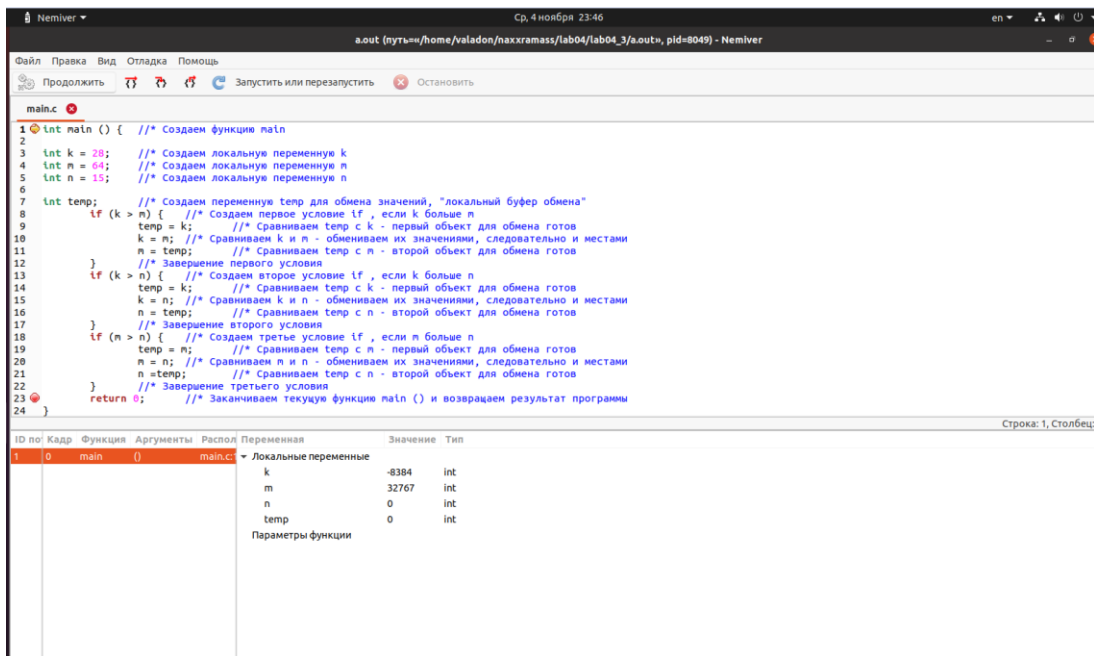
6. Запускаємо відладчик (nemiver), щоб перевірити, як працює наша програма. Щоб запустити програму, треба запустити її через виконувальний файл:

- Відкриваємо “Файл”
- Знаходимо функцію “Завантажити виконувальний файл...”
- Вибираємо програму “a.out” у нашому каталозі

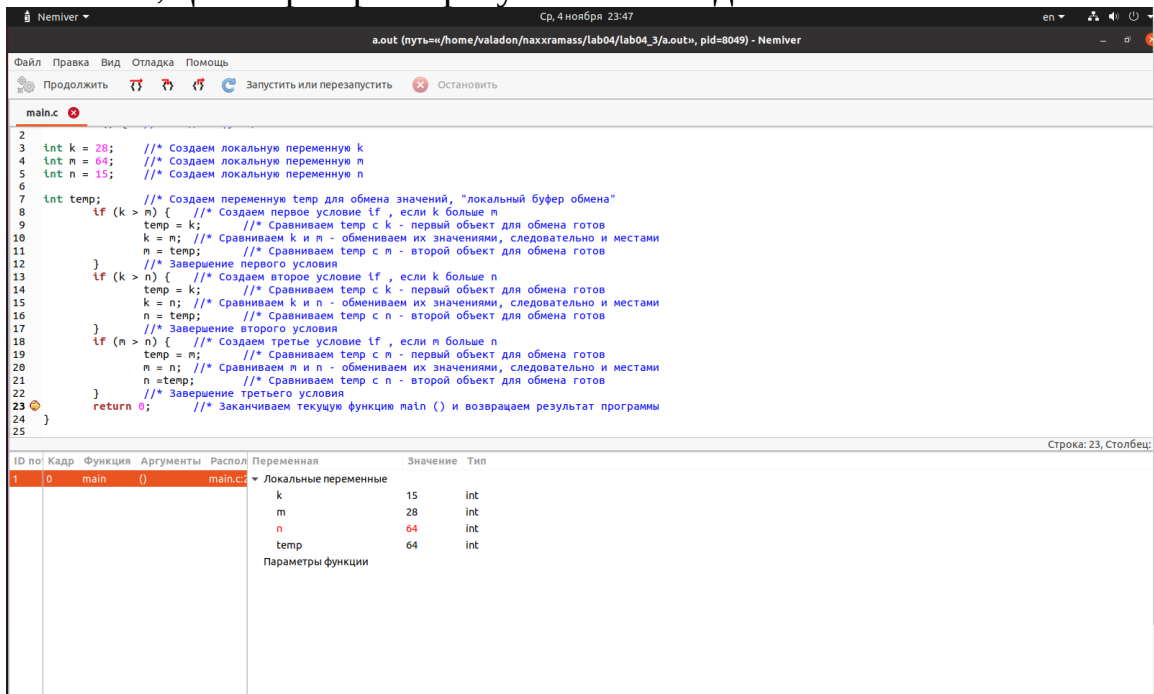


нати”.

Тепер можна побачити програму, яку ми написали раніше:



Н
кції “Step over”, переводимо нашу “стрілку” до рядка `return 0;`
,щоб перевірити результат команди.



у
льтат вірний. Збережемо зміни у нашій директорії на *github*
через команди *git*:
`valadon@valadon-VirtualBox:~/naxxramass $ git add .`
`valadon@valadon-VirtualBox:~/naxxramass $ git status`

```
valadon@valadon-VirtualBox: ~/naxxramass
valadon@valadon-VirtualBox:~/naxxramass$ git add .
valadon@valadon-VirtualBox:~/naxxramass$ git status
На ветке main
Ваша ветка обновлена в соответствии с «origin/main».

Изменения, которые будут включены в коммит:
(use "git restore --staged <file>..." to unstage)
    изменено:      lab03/lab03_good_2/a.out
    новый файл:     lab03/lab03_good_4/Doxyfile
    новый файл:     lab03/lab03_good_4/Makefile
    новый файл:     lab03/lab03_good_4/README.md
    новый файл:     lab03/lab03_good_4/src/main.c
    новый файл:     lab04/lab04_1/a.out
    новый файл:     lab04/lab04_1/main.c
    новый файл:     lab04/lab04_2/a.out
    новый файл:     lab04/lab04_2/main.c
    новый файл:     lab04/lab04_3/a.out
    новый файл:     lab04/lab04_3/main.c
    новый файл:     lab04/lab04_4/a.out
    новый файл:     lab04/lab04_4/main.c
    новый файл:     lab04/lab04_5/a.out
    новый файл:     lab04/lab04_5/main.c
    новый файл:     lab05/lab05_1/a.out
    новый файл:     lab05/lab05_1/main.c
```

```
valadon@valadon-VirtualBox:~/naxxramass $ git commit -m "Create lab04_3"
valadon@valadon-VirtualBox:~/naxxramass $ git push
```

```
valadon@valadon-VirtualBox: ~/naxxramass
valadon@valadon-VirtualBox:~/naxxramass$ git commit -m "Create lab04_3"
[main 88ed904] Create lab04_3
17 files changed, 221 insertions(+)
create mode 100644 lab03/lab03_good_4/Doxyfile
create mode 100644 lab03/lab03_good_4/Makefile
create mode 100644 lab03/lab03_good_4/README.md
create mode 100644 lab03/lab03_good_4/src/main.c
create mode 100755 lab04/lab04_1/a.out
create mode 100644 lab04/lab04_1/main.c
create mode 100755 lab04/lab04_2/a.out
create mode 100644 lab04/lab04_2/main.c
create mode 100755 lab04/lab04_3/a.out
create mode 100644 lab04/lab04_3/main.c
create mode 100755 lab04/lab04_4/a.out
create mode 100644 lab04/lab04_4/main.c
create mode 100755 lab04/lab04_5/a.out
create mode 100644 lab04/lab04_5/main.c
create mode 100755 lab05/lab05_1/a.out
create mode 100644 lab05/lab05_1/main.c
valadon@valadon-VirtualBox:~/naxxramass$ git push
Username for 'https://github.com': Korv3L
Password for 'https://Korv3L@github.com':
Перечисление объектов: 33, готово.
```

3. Варіанти використання

Дана програма являє собою фільтр(товарів, ліків, інші...)

Кожна з цих змінних може являти собою кожну колонку

Наприклад: “немає в наявності” < “найдешевше” < “найдорожче”

4. Висновок

Завдяки цій лабораторній роботі, я навчився робити звичайні програми, що розгалужуються на язику C, у системі Linux, додавати до них коментарі, перевіряти програму на дієздатність у відлагоднику (nemiver), та відправляти зміни у свій репозиторій на github.