

Лабораторна робота №7. Функції.

1 Вимоги

1.1. Розробник 1.1.

- Васюткін Єгор Владиславович
- студент групи КІТ-320
- 15-dec-2020

1.2. Загальне завдання

- Переробити програми, що були розроблені під час лабораторних робіт з тем “Масиви” та “Цикли” таким чином, щоб для обчислення результату використовувалися функції.

1.3. Індивідуальне завдання

- Визначте, чи є задане число простим

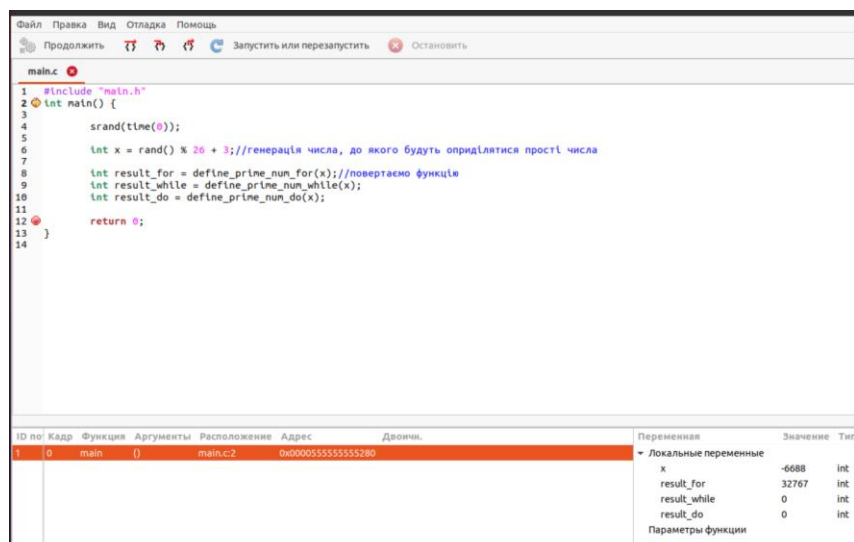
2 Опис роботи

2.1. Функціональне призначення

- Виконання дій програми за допомогою функцій, для спрощення загального завдання програми. Функція допомагає «розбити» велику задачу на підзадачі, у функції main лише зсилатися на їх опис.

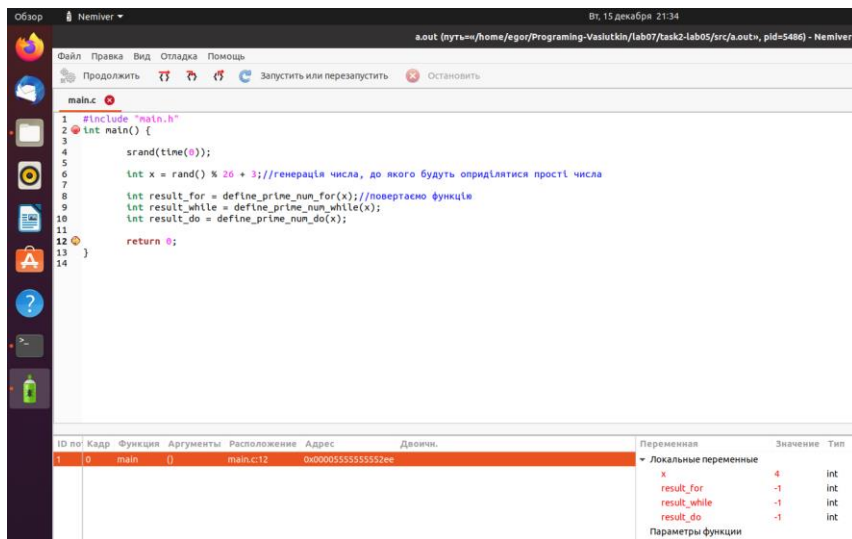
2.2. Варіанти використання Для завдання 2 з лабораторної роботи № 5

- у відладнику netiver викликаємо у функції main три функції, які оприділяють «щасливе» задане число чи ні для циклів for, while do та do while.
- Викликаємо функцію для числа, обчисленого генератором псевдовипадкових чисел rand():

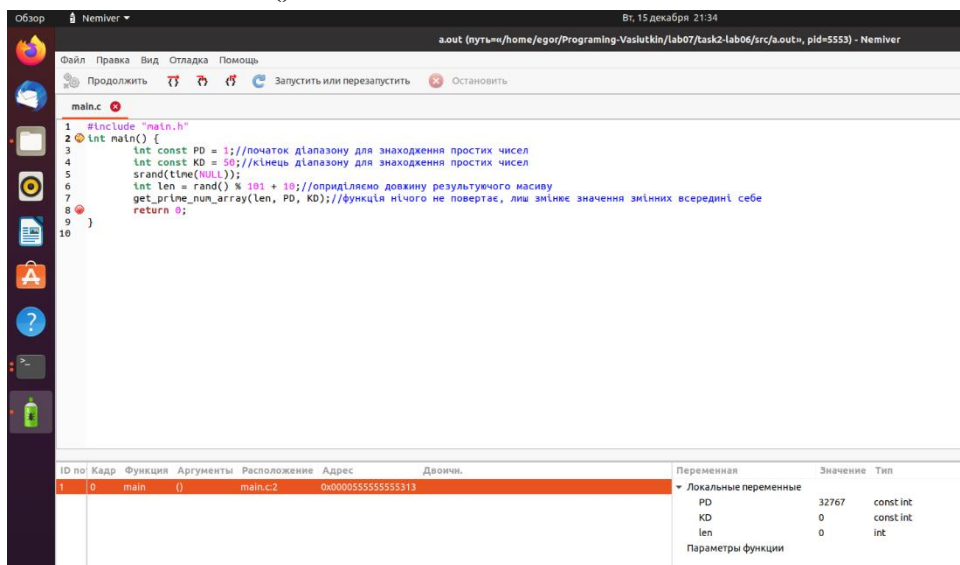


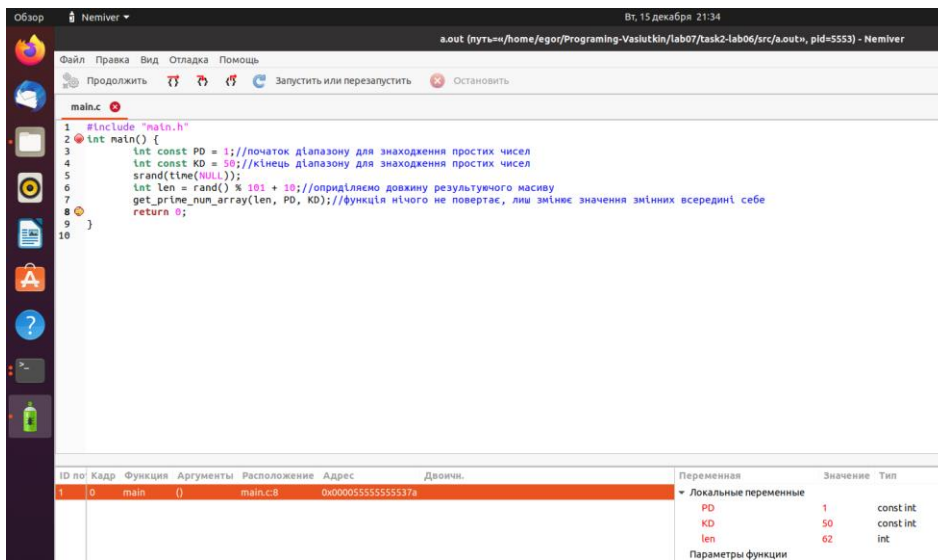
```
1 #include "math.h"
2 int main() {
3
4     srand(time(0));
5
6     int x = rand() % 26 + 3; //генерація числа, до якого будуть оприділятися прості числа
7
8     int result_for = define_prime_num_for(x); //повертаємо функцію
9     int result_while = define_prime_num_while(x);
10    int result_do = define_prime_num_do(x);
11
12    return 0;
13 }
14
```

ID по:	Кадр	Функция	Аргументы	Расположение	Адрес	Двоич.	Переменная	Значение	Тип
1	0	main	()	main.c:2	0x0000555555555280		Локальные переменные		
							x	-6688	int
							result_for	32767	int
							result_while	0	int
							result_do	0	int
							Параметры функции		



- Для завдання 2 з лабораторної роботи № 6
 - у відладнику nemiver за допомогою точок зупину заходимо у функцію, яка заповнює результуючий масив заданим числом, перетвореним у строку
 - задане число генерується за допомогою генератора псевдовипадкових чисел rand():





- Структура проекту:

```

├── doc
│   └── lab07.docx ──
Makefile
├── README.md
├── task1_lab05
│   ├── README.md
│   └── src
│       ├── main.c
│       └── main.h
├── task1_lab06
│   ├── README.md
│   └── src
│       ├── main.c
│       └── main.h
├── task2_lab05
│   ├── README.md
│   └── src
│       ├── main.c
│       └── main.h
├── task2_lab06
│   ├── README.md
│   └── src
│       ├── main.c
│       └── main.h
├── task3_lab05
│   ├── README.md
│   └── src
│       ├── main.c
│       └── main.h
├── task3_lab06
│   ├── README.md
│   └── src
│       └── main.c

```

```
|   └─main.h
├─task4_lab05
|   └─README.md
|   └─src
|       └─main.c
|       └─main.h
├─task4_lab06
|   └─README.md
|   └─src
|       └─main.c
|       └─main.h
├─task5_lab05
|   └─README.md
|   └─src
|       └─main.c
|       └─main.h
├─task5_lab06
|   └─README.md
|   └─src
|       └─main.c
|       └─main.h
└─task6_lab06
    └─README.md
    └─src
└─main.c
└─main.h
```

Висновок

В ході даної лабораторної роботи, я навчився використовувати функції, які повертають і не повертають результат, задля уникання повторів в коді;

«спрощувати» загальне завдання, та для чистоти коду використовував заголовочний файл.