

Міністерство освіти і науки України
Національний технічний університет
«Харківський політехнічний інститут»
Кафедра «Обчислювальна техніка та програмування»

ЗВІТ

Про виконання лабораторної роботи № 6
«Масиви»

Керівник: викладач
Бульба С. С.

Виконавець: студент гр. КІТ-120в
Бабенко А. П.

Харків 2020

Лабораторна робота №6. Масиви

1. Вимоги

1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 15 листопада 2020;

1.2 Загальне завдання

На оцінку “відмінно”. Необхідно виконати усі завдання з даної категорії (проте звіт та відповідні зміни до системи контролю версіями виконуються лише за одним обраним студентом варіантом).

1. Центрувати заданий рядок на площині з із заданим заповнювачем.
Наприклад,
 - заповнювач = “_”,
 - довжина строки = 15,
 - рядок = "Ivanov \0" (6 символів слово “Ivanov”, 8 - пробілів, останній символ = ‘\0’)
 - результат = “____Ivanov____” (4 символи заповнювача, слово “Ivanov”, 4 символи заповнювача, останній символ = ‘\0’)
2. Заповнити масив із заданої кількості елементів простими числами, що не повторюються. Розмір вихідного масиву задати наперед відомим значенням, що може будуть більшим ніж результуюча кількість отриманих елементів.
3. Перетворити число (максимальне значення якого - 9999) в рядок.
Наприклад,
 - 123 – “Сто двадцять три”,
 - 4321 – “Чотири тисячі триста двадцять один”.
4. У заданому тексті знайти кількість слів за умови, що між словами може бути будь-яка кількість пропусків.
5. Дано двовимірний масив з $N \times N$ цілих чисел. Виконати циклічне зрушення елементів рядків масиву в напрямку справа наліво (перший елемент рядка повинен переміститися в її кінець).
6. Дано двовимірний масив з $N \times N$ цілих чисел. Помножити матрицю саму на себе (відповідно до правил множення матриць).

1.3 Індивідуальне завдання

Центрувати заданий рядок на площині із заданим заповнювачем.

2 Виконання роботи

2.1 Створення директорії lab06. Зображено на рис.1.

```
anton@anton-X55VD:~/dev/Programming-repo$ mkdir lab06
```

Рисунок 1 - створення директорії

2.2 Створення файлу з вихідним кодом. Зображено на рис.2.

```
anton@anton-X55VD:~/dev/Programming-repo/lab06$ mkdir src  
anton@anton-X55VD:~/dev/Programming-repo/lab06$ cd src/  
anton@anton-X55VD:~/dev/Programming-repo/lab06/src$ touch task1.c
```

Рисунок 2 - створення файлу з кодом

2.3 Запуск редактора коду. Зображено на рис.3.

```
anton@anton-X55VD:~/dev/Programming-repo/lab06/src$ kate task1.c
```

Рисунок 3 - запуск редактора

2.4 Написання коду і коментарів до нього. Зображено на рис.4.

```

int main()
{
    // Задаем размер массива символов
    #define size 15
    // Задаем заполнитель
    char zap = '_';
    // Задаем массив символов
    char str[size] = "Ivanov      ";
    // Задаем переменные для записи количества пробелов, и символов
    int count_spaces = 0;
    int count_spaces_before_chars = 0;
    int count_chars = 0;
    // Меняем пробелы на заполнитель и считаем кол-во пробелов и символов
    for (int i = 0; i < size - 1; i++)
    {
        if (str[i] == ' ')
        {
            str[i] = zap;
            count_spaces++;
        }
        else
        {
            count_chars++;
        }
    }
    // Считаем кол-во пробелов перед символами
    for (int i = 0; i < size - 1; i++)
    {
        if (str[i] == zap)
        {
            count_spaces_before_chars++;
        }
        else
        {
            break;
        }
    }

    // Сдвигаем слово влево, если пробелы перед словом
    for (int i = 0; i < count_spaces_before_chars; i++)
    {
        for (int j = 0; j < size - 2; j++)
        {
            str[j] = str[j + 1];
        }
        str[size - 2 - i] = zap;
    }
    // Сдвигаем слово на количество пробелов делённое на 2
    for (int i = 0; i < count_spaces / 2; i++)
    {
        for (int j = size - 3; j >= 0; j--)
        {
            str[j + 1] = str[j];
        }
    }
    // Заменяем на заполнитель первые символы кол-во которых равно количеству пробелов пополам
    for (int i = 0; i < count_spaces / 2; i++)
    {
        str[i] = zap;
    }
    return 0;
}

```

Рисунок 4 - код программы

2.5 Компіляція проекту за допомогою команди “make clean prep compile”. Зображено на рис.5.

```
anton@anton-X55VD:~/dev/Programming-repo/lab06$ 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/task1.c -o ./dist/task1.bin
```

Рисунок 5 - компіляція проекту

2.6 Відкрито у відладнику nemiver виконуючий файл task1.bin. Зображено на рис.6.

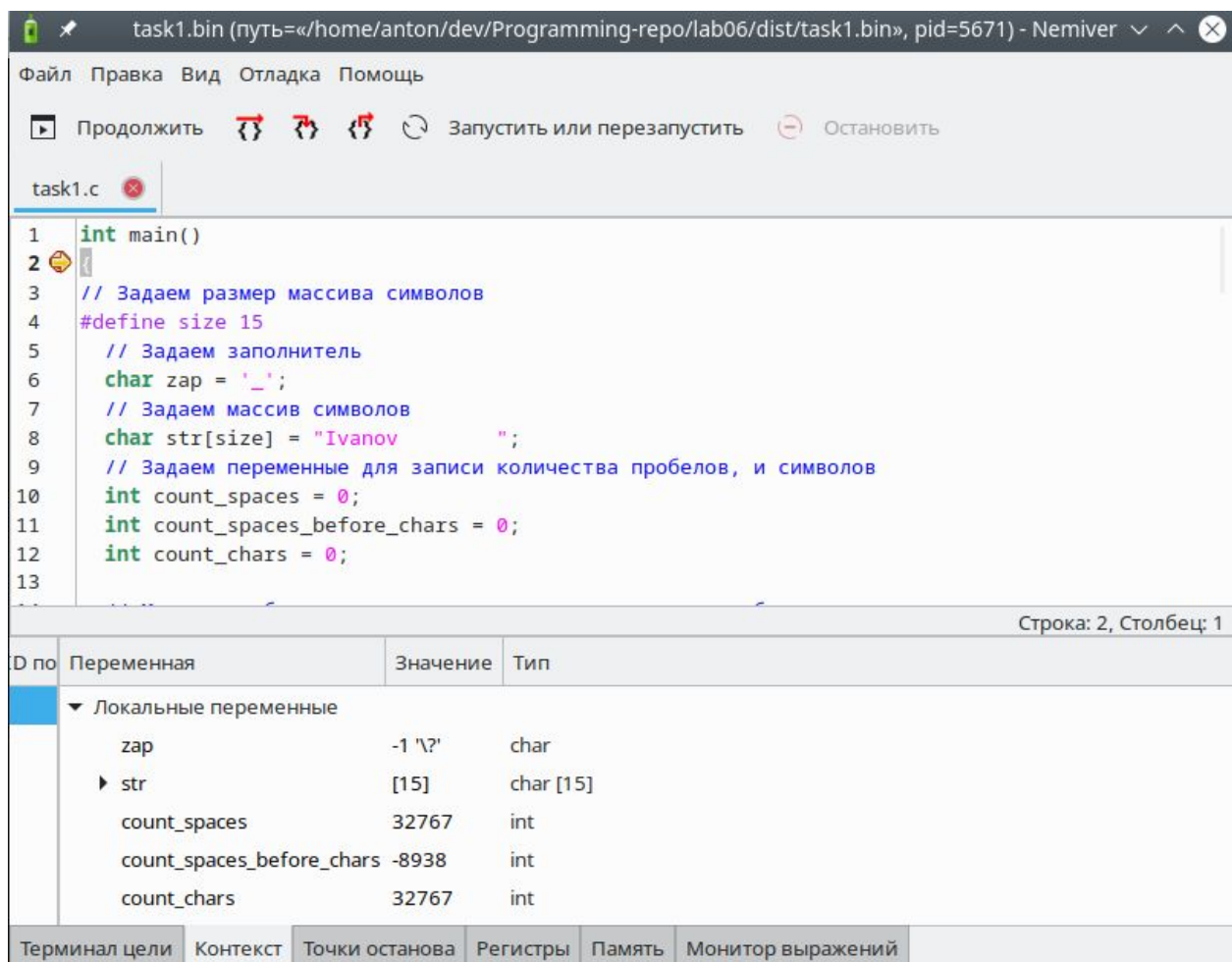


Рисунок 6 - файл у відладнику

2.7 Ставимо точку зупину, проходимо по файлу і бачимо зміну значення в масиві в момент проходження по циклу. Зображено на рис.7.

| Переменная | Значение | Тип |
|------------|----------|-----------|
| ▼ str | [15] | char [15] |
| 0 | 95 '_' | char |
| 1 | 95 '_' | char |
| 2 | 95 '_' | char |
| 3 | 95 '_' | char |
| 4 | 73 'I' | char |
| 5 | 118 'v' | char |
| 6 | 97 'a' | char |
| 7 | 110 'n' | char |
| 8 | 111 'o' | char |
| 9 | 118 'v' | char |
| 10 | 95 '_' | char |
| 11 | 95 '_' | char |
| 12 | 95 '_' | char |
| 13 | 95 '_' | char |
| 14 | 0 '\0' | char |

Рисунок 7 - файл у відлагоднику

2.8 Зафіксовано зміни за допомогою команди “git commit” та завантаження до репозиторію за допомогою команди “git push”. Зображено на рис.8, 9.

```
anton@anton-X55VD:~/dev/Programming-repo$ git commit -m "Added lab06"
[main 492fdc7] Added lab06
```

Рисунок 8 - комміт

```
anton@anton-X55VD:~/dev/Programming-repo$ git push
Перечисление объектов: 13, готово.
Подсчет объектов: 100% (13/13), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (12/12), готово.
Запись объектов: 100% (12/12), 3.21 KiB | 1.07 MiB/s, готово.
Total 12 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Dewwine/Programming-repo.git
50e6a9f..492fdc7 main -> main
```

Рисунок 9 - завантаження змін

2.9 Створення блок-схеми програми. Зображення блок-схеми на рис.10.

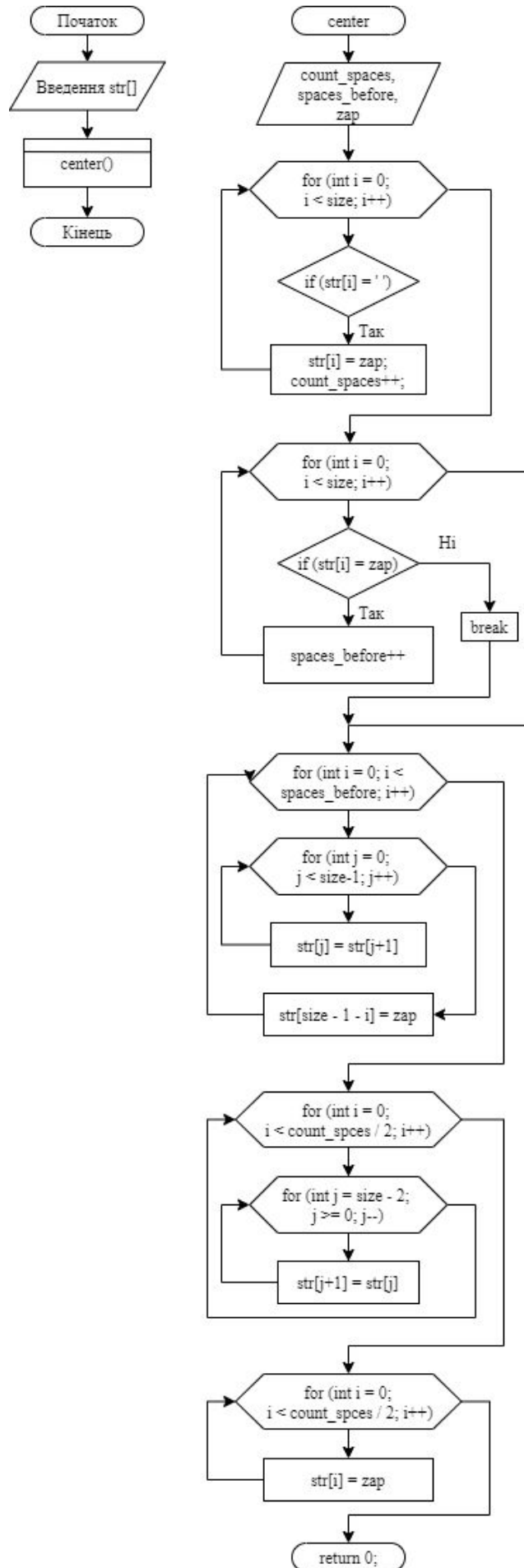


Рисунок 10 - блок-схема

2.10 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 8, 9.

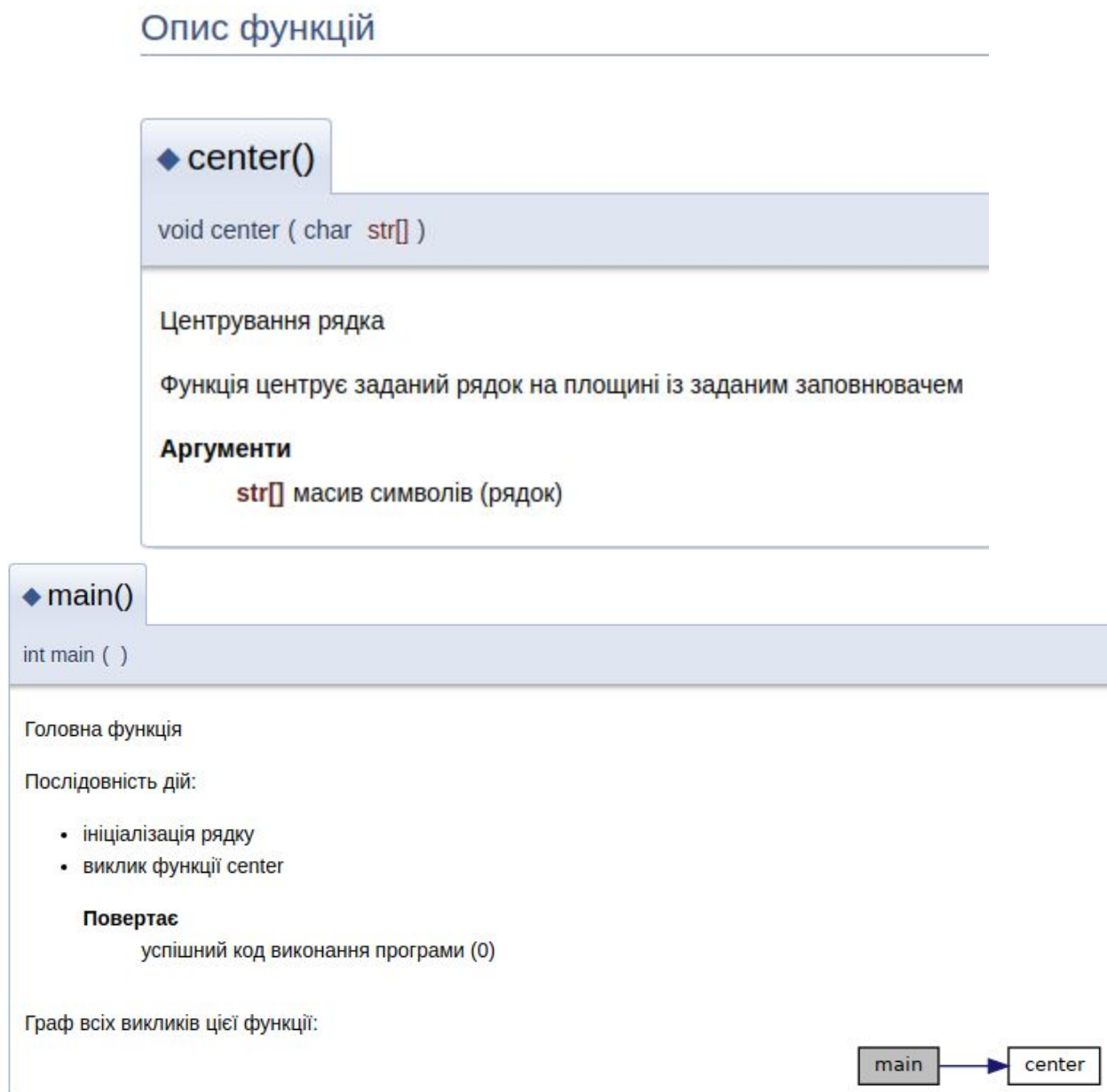


Рисунок 8, 9 — опис функцій

Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм з циклічними конструкціями і розроблено 6 програм, а також створено програму, яка відцентровує слово посеред масиву символів.