

## Лабораторна робота № 6. Масиви

### 1 ВИМОГИ

#### 1.1 Розробник

- Михневич Владислав Вікторович
- Студент 1-го курсу
- Групи КІТ-120а

#### 1.2 Загальне завдання

Центрувати заданий рядок на площині із заданим заповнювачем.

### 2 ОПИС ПРОГРАМИ

#### 2.1 Функціональне призначення

В основу програми покладено від центрування введеного заповнювача в строчці, яка має довжину 15 символів, що декларовані в файлі main.c.

### 3 РОЗРОБКА ПРОГРАМИ ТА ВАРІАНТИ ВИКОРИСТАННЯ

#### 3.1 Структура

Створимо в папці свого репозиторію папку lab\_06, в яку будемо додавати нові файли (див. рис.1).

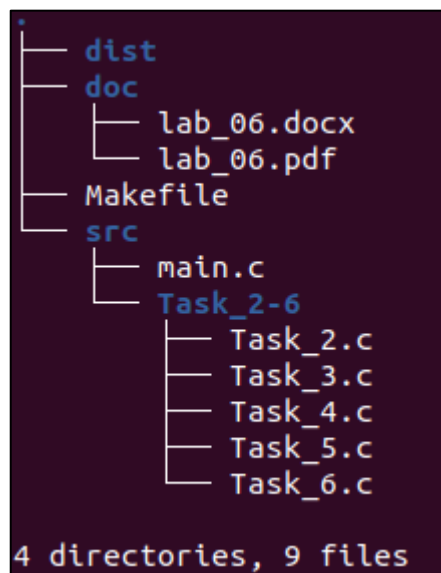


Рисунок 1 – Структура проекту

### 3.2 Початкові дані, типи даних

В нас є константа `SIZE`, так як за умовою задачі чітко зазначений розмір 15. Дано два масиви `word` – який відповідає за задані символи та `container` – який виконує роль прийняття символів слова (не пробілу), за пробіл відповідає змінна `space`, яка має тип `char`. Чотири змінні типу даних `int`: `sz_word` – кількість символів слова, `sz_space` – кількість пробілів з однієї сторони, `a` – кількість символів після дворазового заповнення, створена для межі виконання циклу та `l` – залишок символів (див. рис. 2).

```
#include <stdio.h>
#include <string.h>

#define SIZE 15

int main() {
    char word [SIZE] = "chambo      \0";
    char space = ' ';
    char container [SIZE];
    int sz_word = 0;

    for (int i = 0; word[i] != space; i++){ // количество букв массива word
        sz_word += 1;
    }
    strncpy (container, word, sz_word); // копирование sz_word символов массива word в массив container

    int sz_space = (SIZE - sz_word - 1) / 2; // количество пробелов с одной стороны

    for (int j = 0; j <= sz_space; j++) { // заполнение массива пробелами
        word [j] = space;
    }

    int a = sz_space + sz_word;

    int l = 0;
    for (int m = sz_space; m < a; m++) { // заполнение массива буквами
        word [m] = container [l];
        l += 1;
    }
    for (int k = a; k < SIZE - 1; k++) { // заполнение массива пробелами
        word [k] = space;
    }
    word [14] = '\0';

    return 0;
}
```

Рисунок 2 – Код програми

### 3.3 Додаткова бібліотека

Підключимо бібліотеку `#include <string.h>`, та скористаємося функцією цієї бібліотеки, як копіювання символів масиву `word` в масив `container`:  
`strncpy (container, word, sz_word);`

### 3.4 Робота циклів

Програма має чотири цикли `for` з яких 1-ий відповідає за підрахунок символів слова, яке задано в масиві `word`, наступні три відповідають за

заповнення вихідного рядка, який розбитий на три операції заповнення: пробілами з лівого боку, самим словом та пробілами з правого боку.  
(див. рис. 2).

### 3.4 Компіляція та перевірка в debug

Зробимо компіляцію за допомогою команди `gcc -g main.c -o main.bin`, якщо наша програма написана вірно, то можемо переходити до `nemiver`, побачити значення змінних та процес виконання програми (див. рис. 3).

Переменная	Значение	Тип
▼ Локальные переменные		
▼ word	[15]	char [15]
0	32 ''	char
1	32 ''	char
2	32 ''	char
3	32 ''	char
4	99 'c'	char
5	104 'h'	char
6	97 'a'	char
7	109 'm'	char
8	98 'b'	char
9	111 'o'	char
10	32 ''	char
11	32 ''	char
12	32 ''	char
13	32 ''	char
14	0 '\0'	char
space	32 ''	char
► container	[15]	char [15]
sz_word	6	int
sz_space	4	int
a	10	int
l	6	int

Рисунок 3 – Робота в `nemiver`

### Висновки

На цій лабораторній роботі ми попрацювали з масивами, а саме дізналися як представляються рядки в мові програмування C. Попрацювали з двовірними масивами, та взагалі дізналися скільки вимірів може мати масив. Навчилися звертатися то елементів масиву окремо та отримали навички в оголошенні масиву