Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

**ЗВІТ**

Про виконання лабораторної роботи № 12

«Вступ до показчиків»

Керівник: викладач

Бульба С. С.

Виконавець: студент гр. КІТ-120в

Стегній Б. В.

Харків 2021

**Лабораторна робота № 12. Взаємодія з користувачем шляхом механізму введення/виведення**

**1 Вимоги**

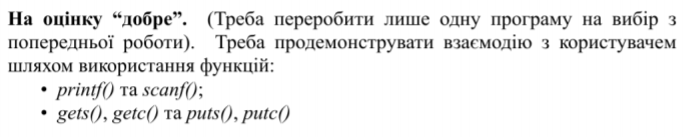
* 1. **Розробник**

• Стегній Богдан Володимирович;

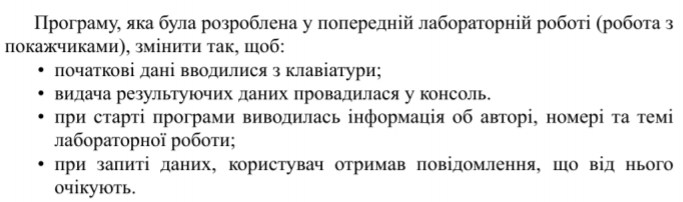
• Студент групи КІТ-120в;

• 13-лют-2021.

* 1. **Загальне завдання**



**1.3 Індивідуальне завдання**



1. **Виконання роботи**
   1. Створив директорію lab12 у репозиторії Programing-repo та заніс туди всі потрібні файли

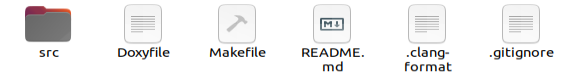


Рисунок 1 – Створення директорії

* 1. Переписав код з одинядцятої лабораторної роботи з використанням функцій введення та виведення данних.

#include <malloc.h>

#include <stdio.h>

#include <time.h>

/\*\* @Filling Функція для заповнення масиву випадковими числами \*/

void Filling(int\*\* Mass, int N) {

/\* Змінна "лічильник" \*/

int z = 0;

// Цикл для заповнення

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

printf("Введите %d элемент массива: ", z+1);

scanf("%d", &Mass[i][j]);

z++;

}

printf("\n");

}

printf("Ваш массив: \n");

/\* Цикл для виведення массиву \*/

for (int i = 0; i < N; i++) {

for (int j = 0; j < N; j++) {

printf("%d\t", \*(\*(Mass + i) + j));

}

printf("\n");

}

}

/\*\* @TransferingDiagonalElements Функція для передачі діагональних елементів в масив для результату\*/

void TransferingDiagonalElements(int\* ResultMass, int\*\* Mass, int N) {

printf("\n");

printf("Диагональные элементы: ");

/\* Цикл для перенесення діагональних елементів в массив для результату \*/

for (int i = 0; i < N; i++) {

\*(ResultMass + i) = \*(\*(Mass + i) + i);

printf("%d\t", \*(ResultMass + i));

}

printf("\n\n");

}

/\*\* @BubbleSort Функція для сортування масиву "бульбашкою"\*/

void BubbleSort(int\* ResultMass, int N) {

/\* Буферна змінна для перестановки елементів масиву \*/

int tmp;

/\* Цикл для "сортування бульбашкою" результуючого масиву \*/

for (int i = 0; i < N - 1; i++) {

for (int j = 0; j < N - i - 1; j++) {

if (\*(ResultMass + j) > \*(ResultMass + j + 1)) {

tmp = \*(ResultMass + j);

\*(ResultMass + j) = \*(ResultMass + j + 1);

\*(ResultMass + j + 1) = tmp;

}

}

}

}

/\*\* @Output Функція для виведення результуючого масиву\*/

void Output(int\* ResultMass, int N) {

printf("Сортировка диагональных элементов: ");

/\* Цикл для виведення елементів масиву \*/

for (int i = 0; i < N; i++) {

printf("%d\t", \*(ResultMass + i));

}

printf("\n\n");

}

/\*\* Основна фунція @main \*/

int main()

{

/\* Створюємо змінні, які знадобляться у подальших процесах введення та виведення \*/

char initial;

char group[4];

char name[100];

char theme[200];

/\* Збір та виведення інформації про користувача \*/

printf("Введите номер и тему лаборатрной работы: ");

gets(theme);

printf("Номер и тема лаборатрной работы: ");

puts(theme);

printf("Введите вашу группу: ");

gets(group);

printf("Ваша группа: ");

puts(group);

printf("Введите ваше имя: ");

gets(name);

printf("Ваше имя: ");

puts(name);

printf("Введите первую букву своего имени: ");

initial = getc (stdin);

printf("Первая буква вашего имени: ");

putc (initial, stdout);

printf("\n");

/\* Визначення розміру масиву \*/

int N;

printf("Введите размер массива: ");

scanf("%d", &N);

for (int i = 0; N < 1; i++) {

if (N < 1) {

printf("Введите положительное число (больше нуля): ");

scanf("%d", &N);

}

}

printf("\n");

/\* Задання динамічного масиву для запису туди результату \*/

int\* ResultMass = (int\*)malloc(N \* sizeof(int));

/\* Задання динамічного двовимірного масиву \*/

int\*\* Mass = (int\*\*)malloc(N \* sizeof(int\*));

for (int i = 0; i < N; i++) {

\*(Mass + i) = (int\*)malloc(N \* sizeof(int));

}

/\*\* Звернення до фунції @Filling \*/

Filling((int\*\*)Mass, N);

/\*\* Звернення до фунції @TransferingDiagonalElements \*/

TransferingDiagonalElements((int\*)ResultMass, (int\*\*)Mass, N);

/\*\* Звернення до фунції @BubbleSort \*/

BubbleSort((int\*)ResultMass, N);

/\*\* Звернення до фунції @Output \*/

Output((int\*)ResultMass, N);

return 0;

}

**2.3** Зробив компіляцію проекту та оцінив його парцездатність.

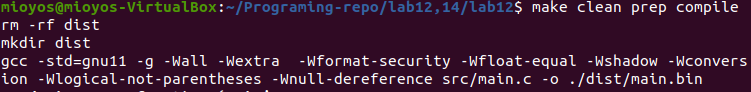


Рисунок 2 – компіляція

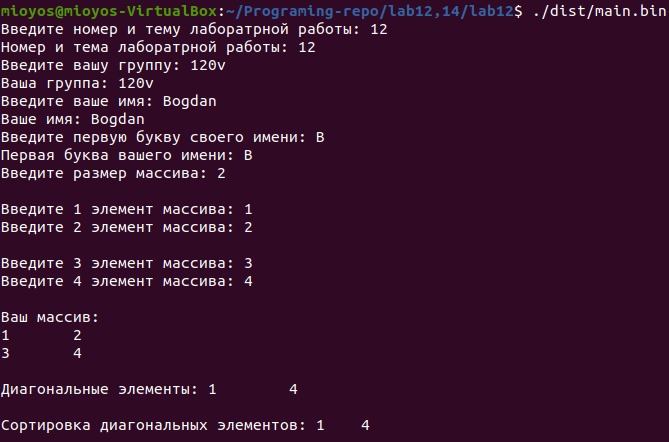


Рисунок 3 – тест

**2.6** Згенерував html файл за допомогою команди make doxy



Рисунок 4 – make doxy

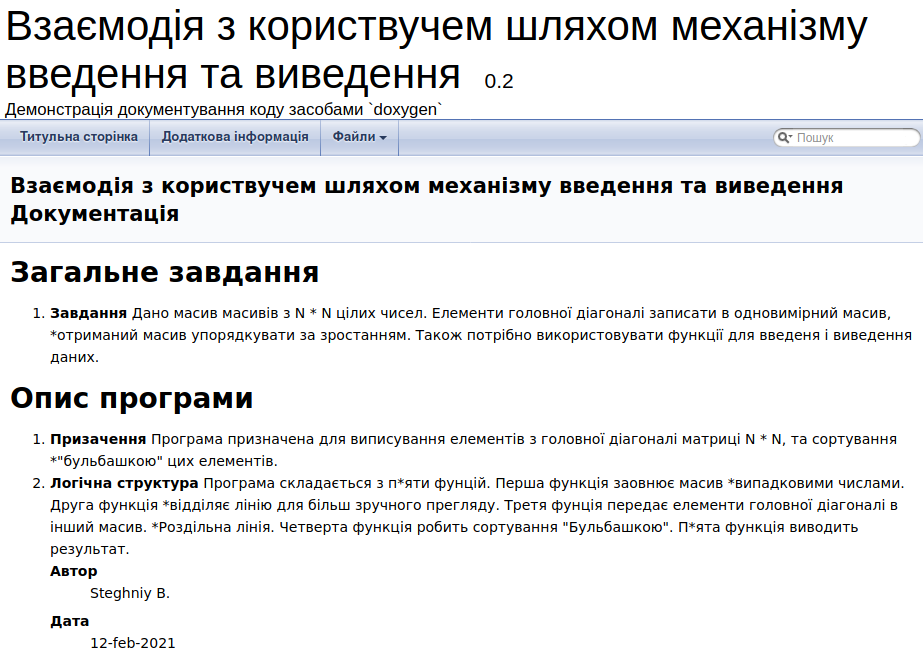


Рисунок 5 – Doxygen

**2.7** Усі зміни були додані, зафіксовані та завантажені до онлайн-репозиторію Programing-repo за допомогою команд git add, git commit, git push.

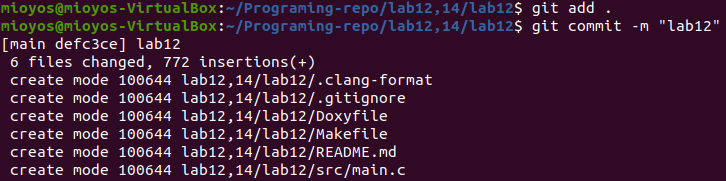


Рисунок 6 – git add, git commit

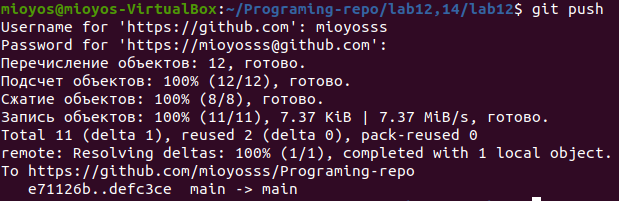


Рисунок 7 – git push

**2.8** Зробив блок-схему для програми.

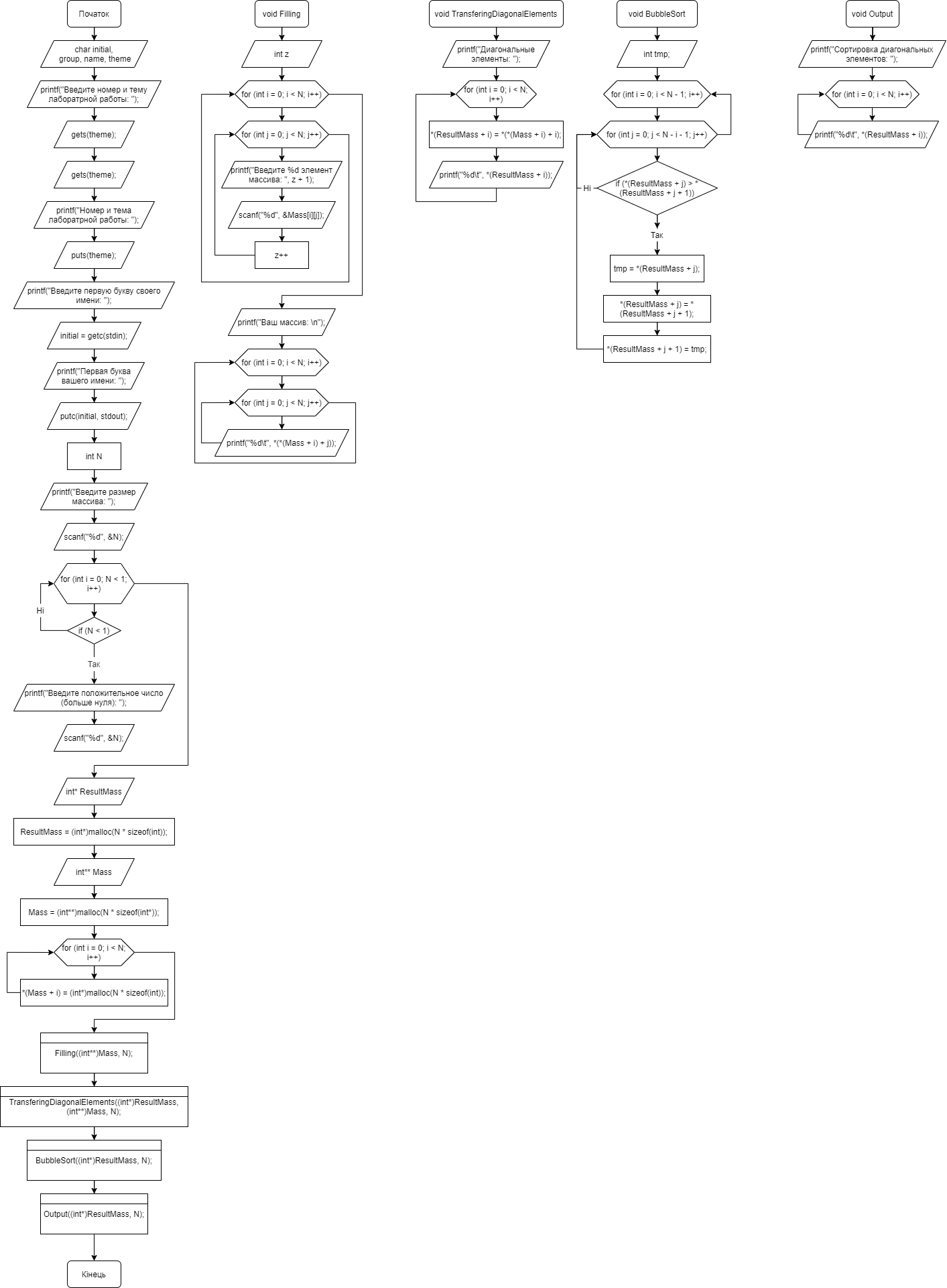


Рисунок 8 – блок-схема

**Висновки**

При виконанні лабораторної роботи № 12 «Взаємодія з користувачем шляхом механізму введення/виведення», переробив програму таким чином, щоб у її виконанні були задіяні механізми введення та виведення. Були розвинуті нові навички у роботі з середовищем Linux.