

Лабораторна робота №10. «Вступ до документації проекту»

Виконав: ст. гр. КІТ-120А Старовойтов Н.А.
Перевірив: Челаєк В.В.

Мета: Отримати навички роботи у Markdown та оформлення doc-файлу згідно з ГОСТом.

Індивідуальне завдання

Робота на оцінку "відмінно". * Завдання №1: «Визначити найбільший спільний дільник для двох заданих чисел». * Завдання №2: «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)». * Завдання №3: «Реалізувати функцію, що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію з варіативною кількістю аргументів».

Опис програми

Алгоритм вирішення поставленої задачі

Для вирішення Завдання №1 використовується програмна реалізація алгоритма Євкліда. Функція Find_gcf приймає два числа та повертає їх НСД. Викликається функція двічі – з заданими та згенерованими випадково числами.

◆ Find_gcf()

```
int Find_gcf ( int a,
               int b
             )
```

Знаходження НСД двох чисел

Аргументи

- a** - Перше число
- b** - Друге число

Повертає

result - НСД двох чисел

Рисунок 1 — Функція Find_gcf

Для вирішення Завдання №2 використовується покроковий зсув кожного рядка разом з одночасним записом першого елемента рядка у масив temp. Після зсуву останньому елементу рядка надається значення, що було збережено у масиві temp. Операція повторюється N разів. Всі ці дії виконуються у функції Shift_matrix_right_1. Для заповнення масиву було використано генератор псевдовипадкових чисел. Результат зберігається у тому ж масиві, що й первісні дані.

◆ Shift_matrix_right_1()

```
int Shift_matrix_right_1 ( int array[] )
```

Циклічний зсув (зрушення) кожної строки матриці на 1 праворуч

Аргументи

- array** - Матриця

Повертає

0 - Код успішного виконання функції

Рисунок 2 — Функція Shift_matrix_right_1

Для вирішення Завдання №3 використовується функція Num_of_ordered_pairs з варіативною кількістю аргументів. Вона приймає число – кількість наступних аргументів, та, власне, ці аргументи. Повертає число – кількість пар чисел, що відповідають умові задачі. Викликається функція двічі – з заданими та згенерованими випадково числами.

◆ Num_of_ordered_pairs()

```
int Num_of_ordered_pairs ( int size,  
    ...  
    )
```

Знаходження серед заданої послідовності чисел кількості таких пар, у котрих перше число менше наступного

Аргументи

size - Кількість чисел у послідовності

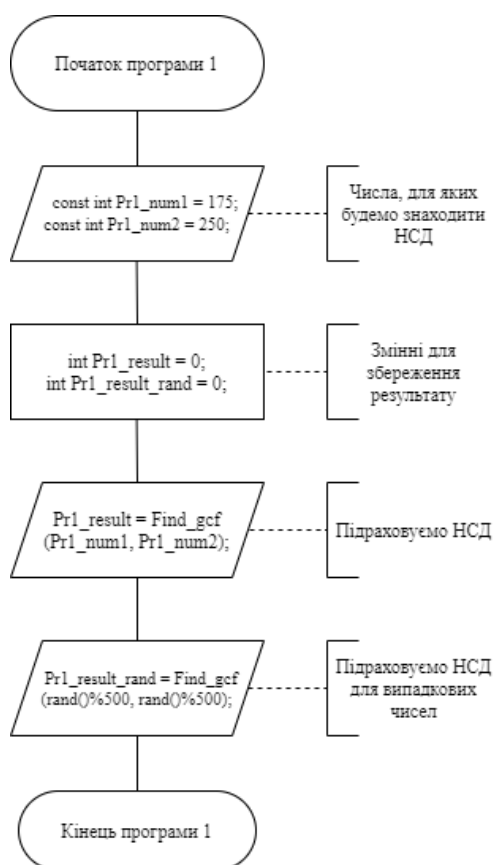
... - Числа послідовності

Повертає

result - Кількість пар, що відповідають заданим параметрам

Рисунок 3 – Функція Num_of_ordered_pairs

Схеми алгоритмів функцій



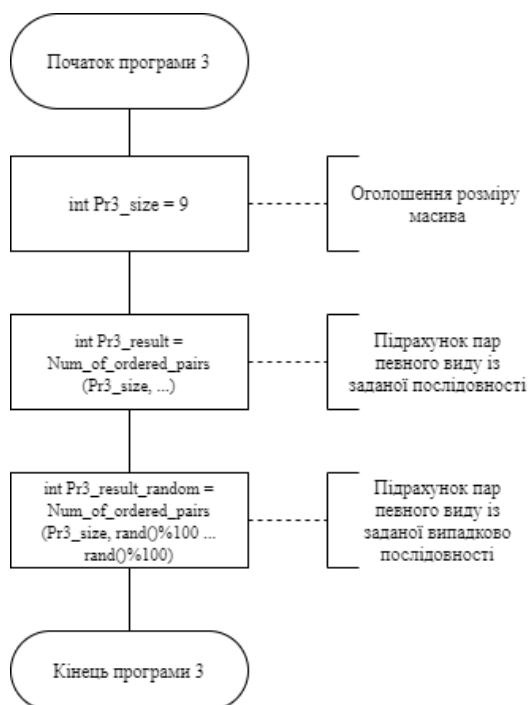
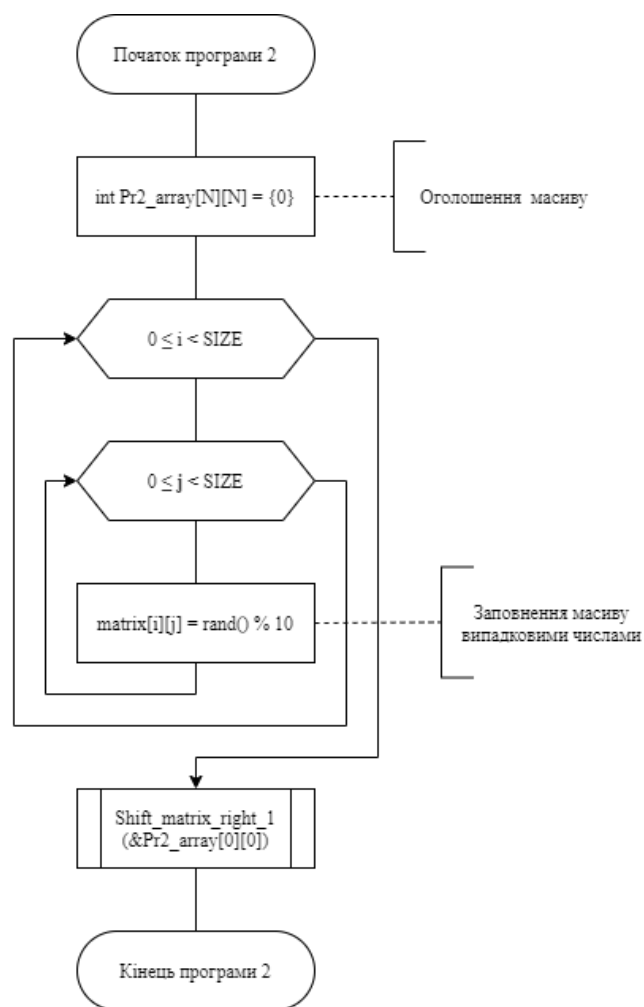


Рисунок 4 – Блок-схема функції `taip`, яка складається з трох програм

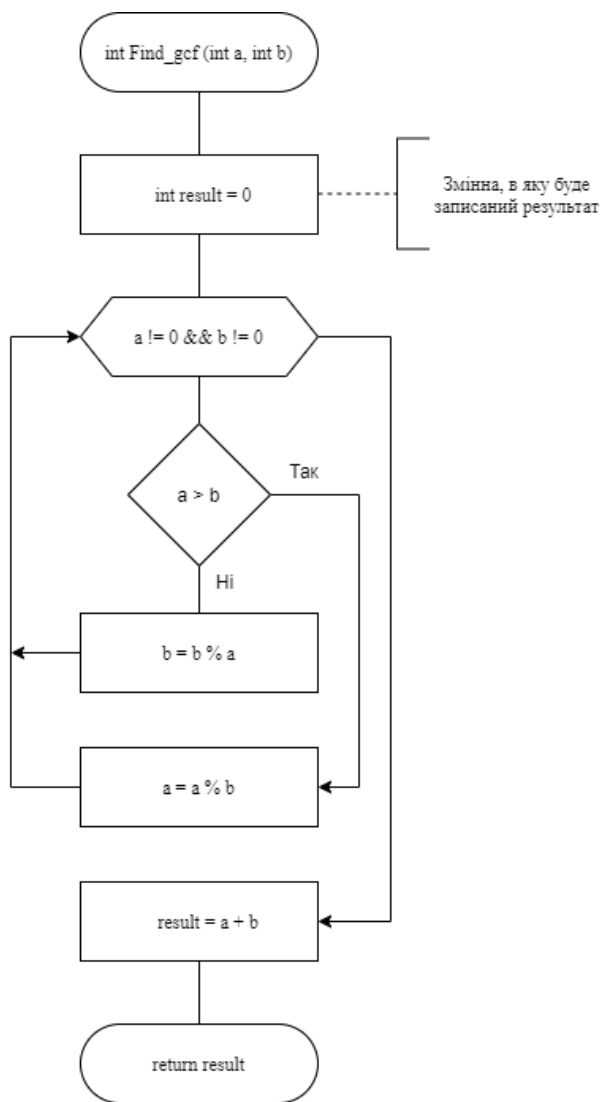


Рисунок 5 – Блок-схема функції Find_gcf

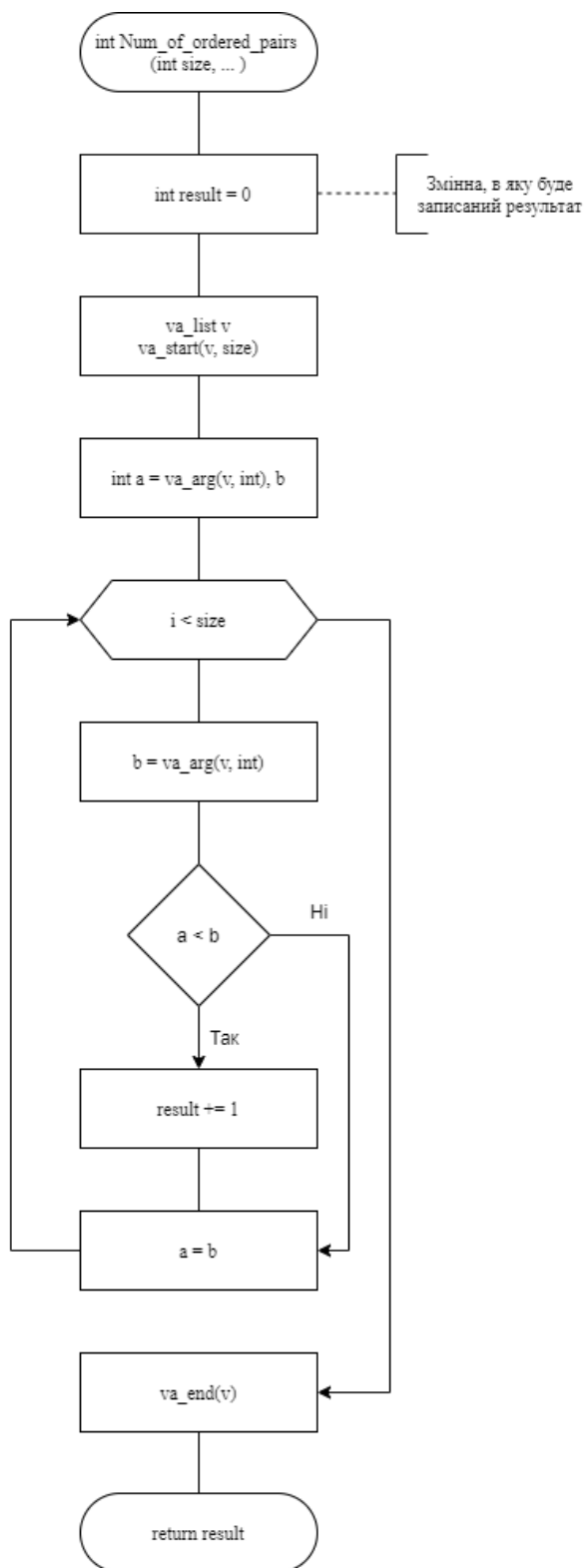


Рисунок 6 – Блок-схема функції `Num_of_ordered_pairs`

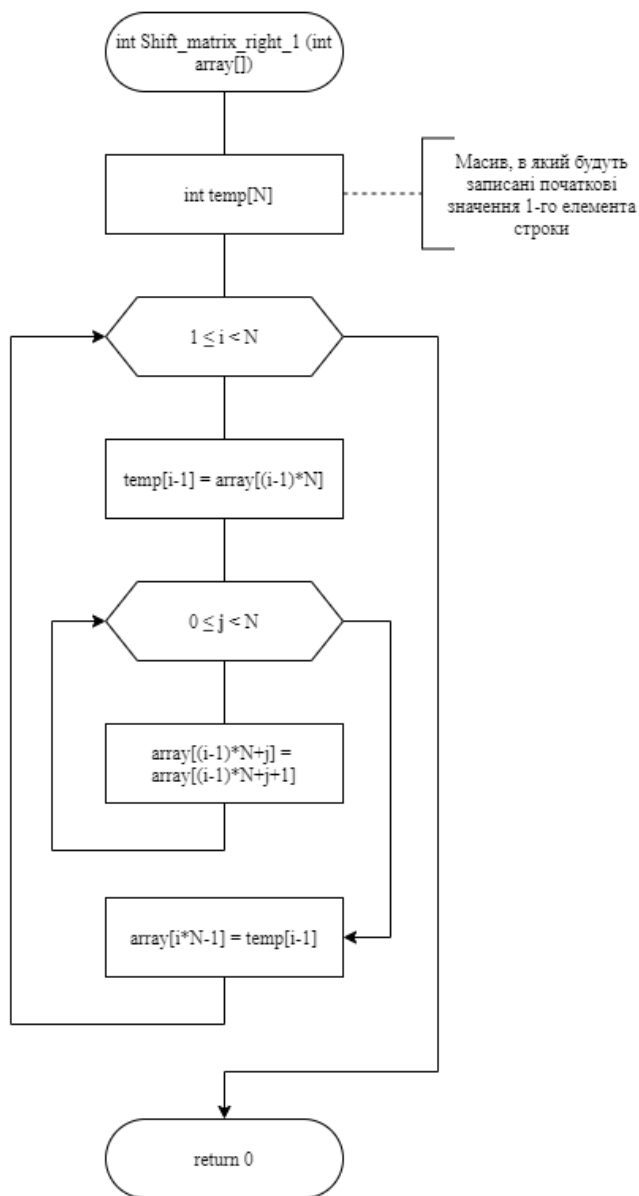


Рисунок 7 – Блок-схема функції Shift_matrix_right_1

Текст програми

```

#include <stdlib.h>
#include <stdarg.h>
#include <time.h>

int Find_gcf (int a, int b){
    int result = 0;
    while (a != 0 && b != 0){
        if (a > b) a = a % b;
        else b = b % a;
    }
    result = a + b;
    return result;
}

int Shift_matrix_right_1 (int array[]) {
    int temp[N];
    for (int i = 1; i <= N; i++) {
        temp[i-1] = array[(i-1)*N];
        for (int j = 0; j < N; j++) {
            array[(i-1)*N+j] = array[(i-1)*N+j+1];
        }
        array[i*N-1] = temp[i-1];
    }
    return 0;
}

```

```

}

int Num_of_ordered_pairs (int size, ... )
{
    int result = 0;
    int i = 1;
    va_list v;
    va_start(v, size);
    int a = va_arg(v, int), b;
    while (i < size){
        b = va_arg(v, int);
        if (a < b) result += 1;
        a = b;
        i++;
    }
    va_end(v);
    return result;
}

int main(){
    ///// Program 1 /////
    srand(time(NULL));
    const int Pr1_num1 = 175;
    const int Pr1_num2 = 250;
    int Pr1_result = 0, Pr1_result_rand = 0;
    Pr1_result = Calculation1 (Pr1_num1, Pr1_num2);
    Pr1_result_rand = Calculation1 (rand()%500, rand()%500);

    ///// Program 2 /////
    int Pr2_array[N][N] = {0};
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            Pr2_array[i][j] = rand() % 10;
        }
    }
    Calculation2 (&Pr2_array[0][0]);

    ///// Program 3 /////
    int Pr3_size = 9;
    int Pr3_result = Calculation3 (Pr3_size, 2, 2, 8, 7, 9, 3, 12, 5, 99);
    int Pr3_result_random = Calculation3 (Pr3_size, rand()%100, rand()%100, rand()%100, rand()%100, rand()%100, rand()%100, rand()%100, rand()%100);

    return 0;
}

```

Результати роботи програми

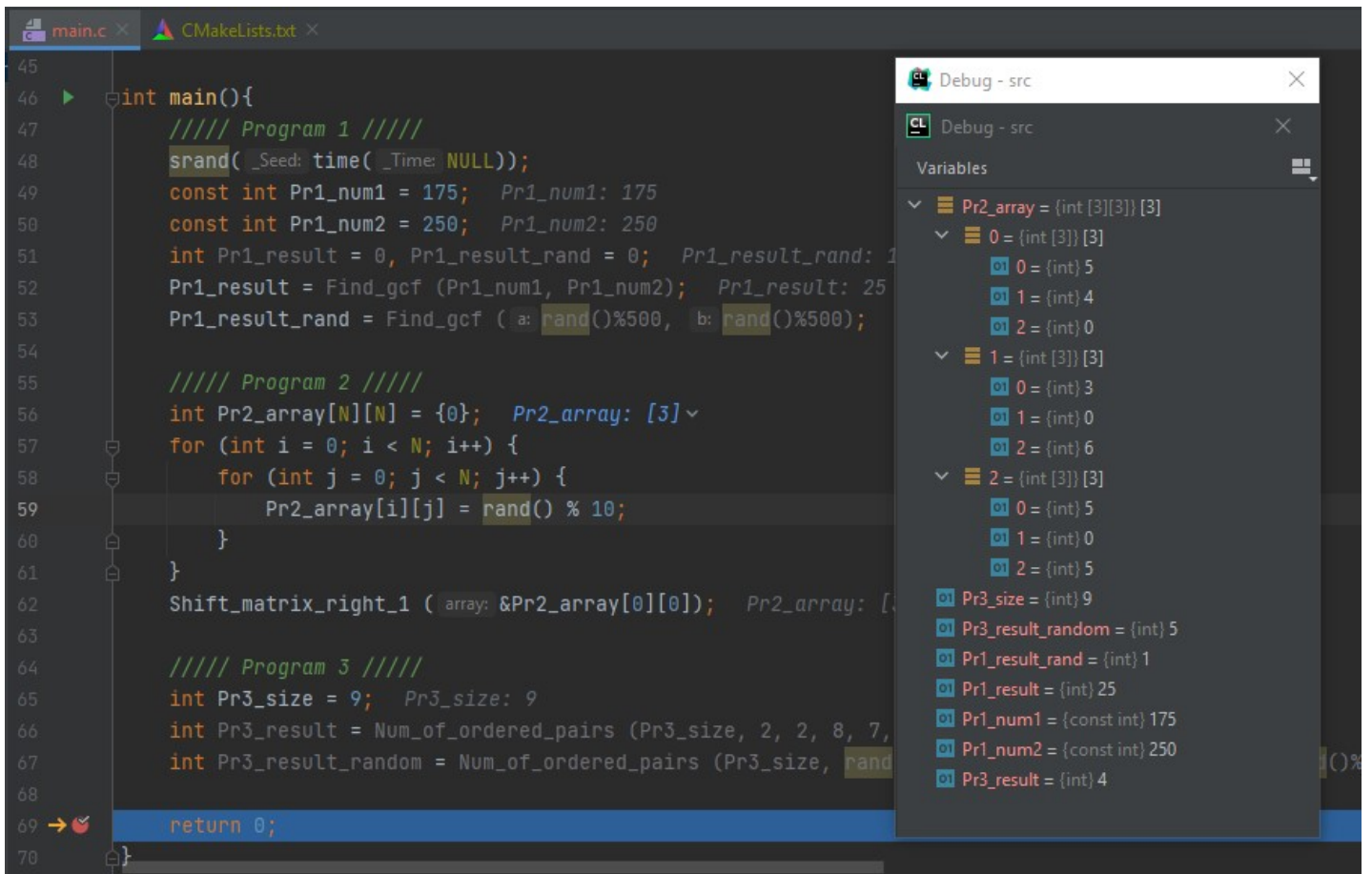


Рисунок 8 – Результат успішного виконання програми

Висновки

Під час виконання даної лабораторної роботи було отримано навички роботи у Markdown та оформлення дос-файлу згідно з ГОСТом.