

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

ЗВІТ

Про виконання лабораторної роботи № 11
«Вступ до показчиків»

Керівник: викладач
Бульба С. С.

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

Харків 2020

Лабораторна робота №11. Вступ до показчиків

1. Вимоги

1.1 Розробник

- Бабенко Антон Павлович;
- Студент групи КІТ-120в;
- 12 грудня 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. Дано масив з N цілих чисел. Визначити, чи є в масиві елементи, що повторюються; якщо такі є, то створити масив, в якому вказати, скільки разів які елементи повторюються. Таким чином, в результуючому масиві кожен непарний елемент - число, що повторюються; кожен парний елемент - кількість повторювань.
2. Дано масив з N цілих чисел. Знайти безперервну послідовність позитивних чисел у вхідному масиві, сума елементів якої максимальна, та переписати їх у вихідний масив.
3. Дано масив з N речовинних чисел. Підрахувати кількість ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями. Максимальну ділянку переписати у інший масив.
4. Дано масив масивів з $N * N$ цілих чисел. Елементи головної діагоналі записати в одновимірний масив, отриманий масив упорядкувати за зростанням.

1.3 Індивідуальне завдання

Дано масив з N цілих чисел. Визначити, чи є в масиві елементи, що повторюються. В результуючому масиві кожен непарний елемент непарний елемент - число, що повторюється, а кожний парний - кількість повторювань.

1.4 Функціональне призначення

Програма призначена для запису в результуючий масив числа і кількості його повторювань в початковому масиві.

2 Виконання роботи

2.1 Створення директорії lab11. Зображено на рис.1.

```
anton@anton-X55VD:~/dev/Programming-repo$ mkdir lab11
```

Рисунок 1 – створення директорії

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

```
anton@anton-X55VD:~/dev/Programming-repo/lab11$ mkdir src  
anton@anton-X55VD:~/dev/Programming-repo/lab11$ touch src/main.c
```

Рисунок 2 – створення файлу з кодом

2.3 Запуск редактора коду. Зображено на рис.3.

```
anton@anton-X55VD:~/dev/Programming-repo/lab11$ kate src/main.c
```

Рисунок 3 – запуск редактора

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

```

void fill(int *, int);
void repeat(int *, int);

int main()
{
    srand(time(0));
    int N = 5;
    int * arr = (int*)calloc(N, sizeof(int));

    fill(arr, N);
    print(arr, N);
    printf("\n");
    repeat(arr, N);

    free(arr);

    return 0;
}

void fill(int * arr, int N) {
    for (int i = 0; i < N; i++)
    {
        *(arr + i) = rand() % 10 + 1;
    }
}

void repeat(int * arr, int N) {
    int count = 1;
    int x = 0;
    int * res = (int*)calloc(N * 2, sizeof(int));

    for (int i = 0; i < N; i++) {
        for (int j = i+1; j < N; j++) {
            if (*(arr + i) == *(arr + j)) {
                *(res + x) = *(arr + j);
                count++;
                *(arr + j) = 0;
            }
        }
        if (*(res + x) != 0) {
            *(res + x + 1) = count;
            x += 2;
        }
        count = 1;
    }
}

```

Рисунок 4 – код програми

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

```
anton@anton-X55VD:~/dev/Programming-repo/lab11/01$ make clean prep compile
```

Рисунок 5 – компіляція проекту

2.6 Відкрито у відлагоднику nemiver виконуючий файл main.bin. Зображено на рис.6.

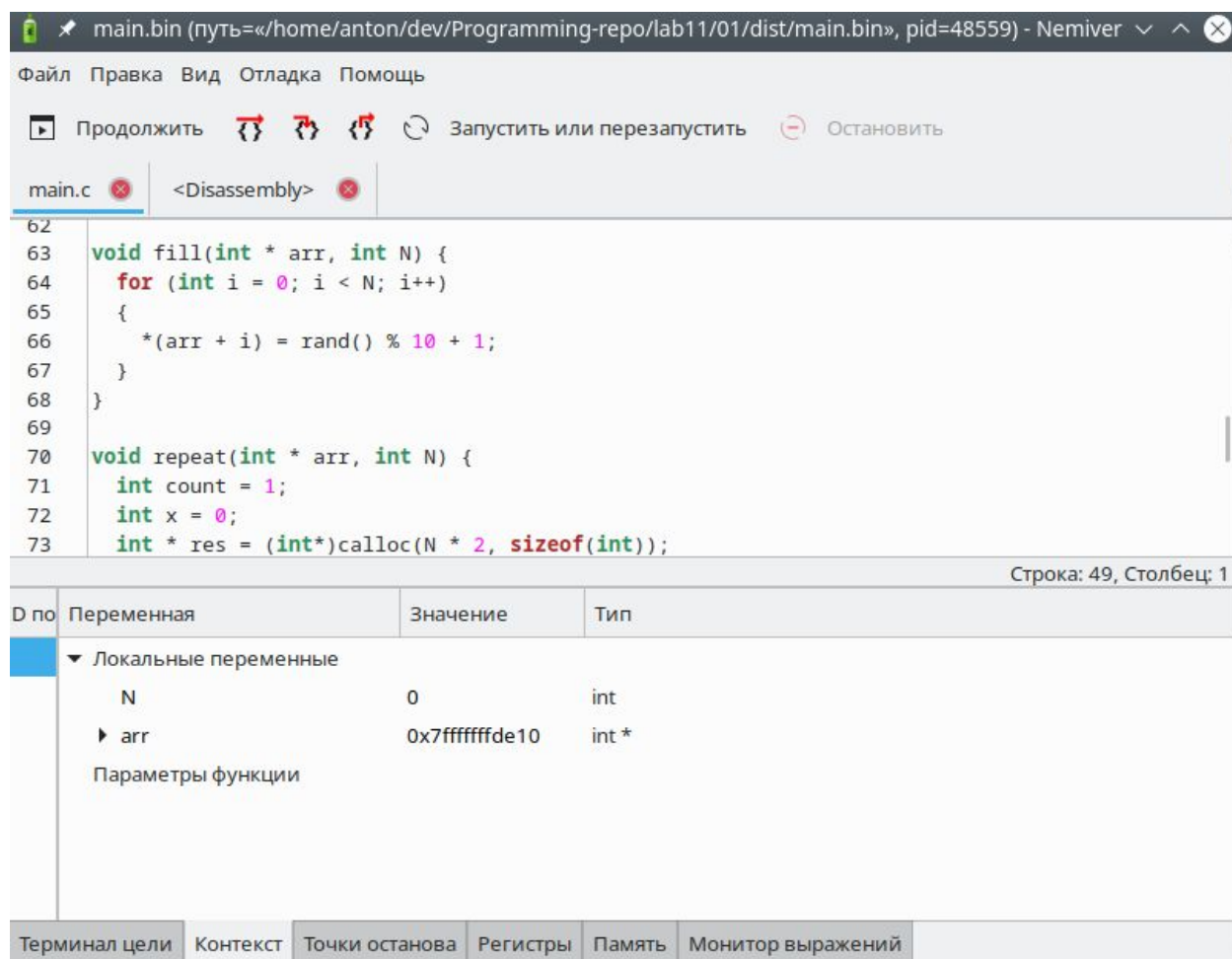


Рисунок 6 – файл у відлагоднику

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

Строка: 79, Столбец: 1			
D по	Переменная	Значение	Тип
▼ Локальные переменные			
	count	1	int
	x	0	int
	▼ res	0x4056d0	int *
	*res	4	int
	i	1	int
	j	4	int
<div> <div>Терминал цели</div> <div>Контекст</div> <div>Точки останова</div> <div>Регистры</div> <div>Память</div> <div>Монитор выражений</div> </div>			

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 lab11"
```

Рисунок 8 – комміт

```
anton@anton-X55VD:~/dev/Programming-repo$ git push
Перечисление объектов: 23, готово.
Подсчет объектов: 100% (23/23), готово.
При сжатии изменений используется до 2 потоков
Сжатие объектов: 100% (18/18), готово.
Запись объектов: 100% (22/22), 4.85 KiB | 709.00 KiB/s, готово.
Total 22 (delta 5), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (5/5), completed with 1 local object.
To https://github.com/Dewwine/Programming-repo.git
570b9bc..4d29e2e  main -> main
```

Рисунок 9 – завантаження змін

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

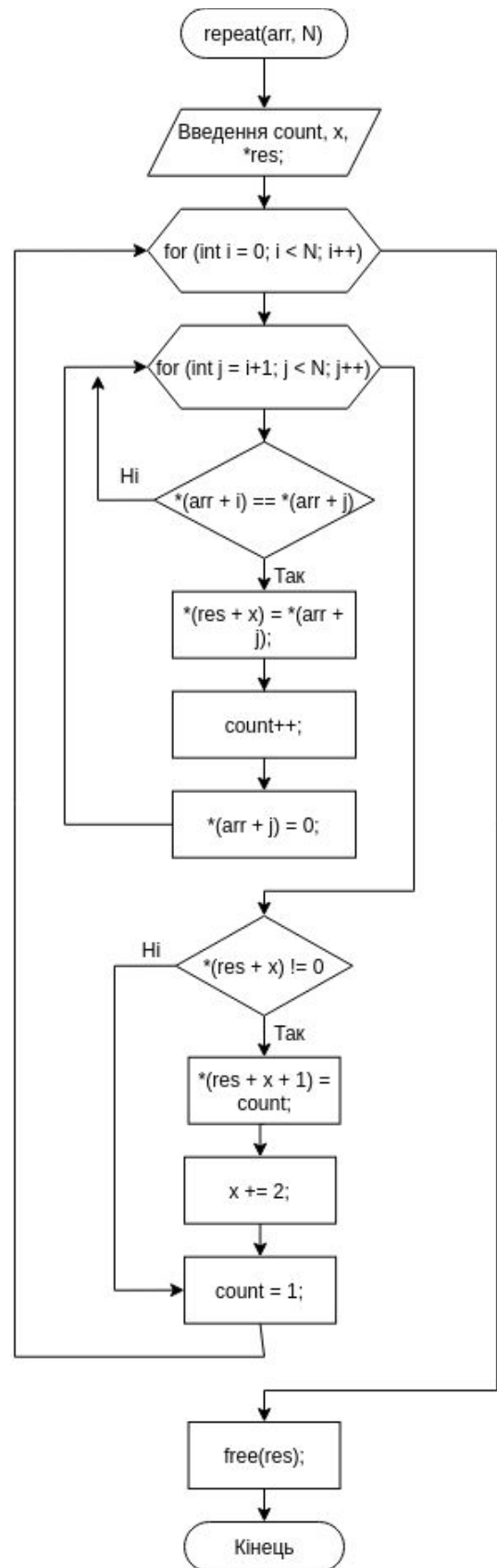
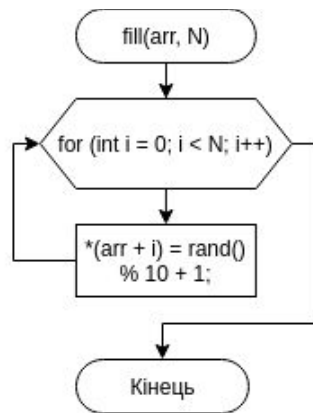
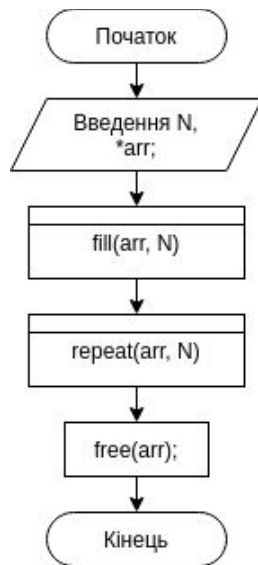


Рисунок 10 – блок-схема

2.10 Виконання опису функцій. Зображення опису зі сторінок документації doxygen на рис. 11, 12, 13.

◆ fill()

```
void fill ( int * arr,  
           int  N  
           )
```

Заповнення масиву числами

Функція, що заповнює масив псевдовипадковими числами

Аргументи

arr масив чисел

N кількість елементів масиву

◆ repeat()

```
void repeat ( int * arr,  
             int  N  
             )
```

Заповнення масиву числами що повторюються

Функція, що шукає числа які повторюються і кількість повторювань

Аргументи

arr заповнений масив чисел

N кількість елементів масиву

◆ main()

```
int main ( )
```

Головна функція

Послідовність дій:

- задання кількості елементів масиву
- ініціалізація рядку
- виклик функції fill
- виклик функції repeat

Повертає

успішний код виконання програми (0)

Граф всіх викликів цієї функції:

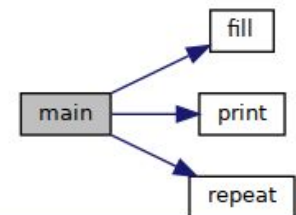


Рисунок 11, 12, 13 – опис функцій

Висновки

При виконанні даної лабораторної роботи було набуто навичок розробки програм з показниками і розроблено 4 програми, а також створено програму, яка записує в результуючий масив число із початкового масива що повторюється і кількість його повторень.