

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

КАФЕДРА «Обчислювальної техніки та програмування»

Лабораторна робота №11

Тема: «Вказівник»

Виконала :

ст. гр. КІТ-120В, Семенченко Вікторія;

Перевірив:

Бульба С.С.

Харків 2020

Мета: розв'язати завдання з категорії «добре», використовуючи вказівники.

Звіт виконується за варіантом, обчисленим за формулою :

$$N_t = ((N_j - 1) \% C) + 1,$$

де:

- N_j – номер студента у журналі групи;
- C – кількість варіантів у лабораторній роботі (для розраховуємого рівня);
- $\%$ – Ділення з остачею.

Номер мого варіанту 2.

У завданні №2 потрібно розробити програму, яка рахує кількість парних додатніх чисел у кожному рядку.

1. Написання коду програми

```
i8 */
i9 void filling(int arr[][N]){
i10     srand(time(0));
i11     for(int i=0;i<N;i++){
i12         for(int j=0;j<N;j++){
i13             (*(arr+i)+j)=rand()%20-10;
i14         }
i15     }
i16 }
i17
i18 /**
i19  * @function cycle
i20  *
i21  * Функція для визначення всіх парних додатніх чисел
i22  * Послідовність дій:
i23  * - оголошення змінних
i24  *   @param a кількість парних додатніх чисел
i25  * - Цикл для перевірки кожного елементу двовимірного масиву:
i26  *   - Якщо число додатне і парне - а++ і число записується у одновимірний масив
i27  */
i28 void cycle(int arr[][N], int arr_res[]){
i29     int a=0;
i30     for(int i=0;i<N;i++){
i31         for(int j=0;j<N;j++){
i32             if(*(arr+i)+j)%2==0&&*(arr+i)+j>0){
i33                 *(arr_res+a+1)=*(arr+i)+j;
i34                 a++;
i35                 *(arr_res)=a;
i36             }
i37         }
i38     }
i39     for(int i=0;*(arr_res+i)!=0;i++){
i40         printf("%d ", *(arr_res+i));
i41     }
i42 }
```

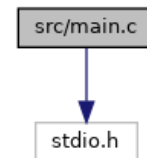
Рисунок 1.1 – код програми

Файл main.c

Файл з демонстрацією роботи структури та методів оперування ним. [Детальніше...](#)

```
#include "stdio.h"
```

Діаграма включених заголовочних файлів для main.c:



Макровизначення

```
#define N 5
```

Функції

```
void cycle (int arr[][N], int arr_res[])  
@function cycle Детальніше...
```

```
void filling (int arr[][N])  
@function filling Детальніше...
```

```
int main ()  
Головна функція. Детальніше...
```

Рисунок 1.2 – Doxygen документація

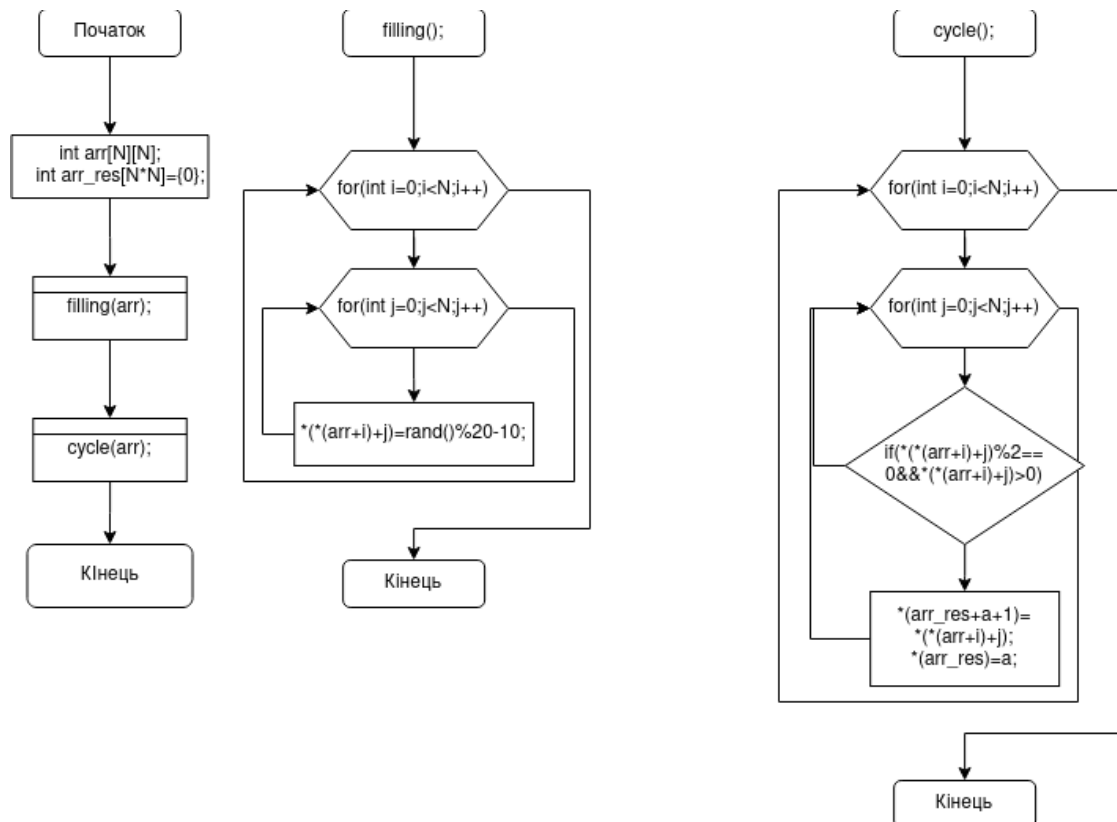


Рисунок 1.4 – блок-схема алгоритму

Відповіді на питання :

1. Треба вказати тип даних, що використовується покажчик, після цього поставити символ «*», вказати назву покажчика, після чого вказати адресу на нульовий елемент масиву.
2. Над покажчиками можна виконувати наступні операції:
 - Присвоєння.
 - Визначення значення.
 - Одержання адреси покажчика.
 - Збільшення покажчика.
 - Різниця покажчиків.
 - Порівняння (за адресою).
 - Присвоєння константи NULL
3. За допомогою оператора free(назва масиву);
4. Якщо масив фіксований, то виділяти пам'ять ми будемо лише один раз, коли вказуємо розмір масиву. Якщо ж у нас масив має змінний

розмір, то потребується комбінація операторів `calloc/malloc` та `free`(назва масиву) для видалення масиву з пам'яті.

5. Щоб створити контроль за витоків пам'яті треба видалити масив після того, як він був використаний і більше не потрібний. Якщо це багатомірний масив, треба спочатку видалити масиви найменшого порядку, після чого перейти на видалення елементів наступного порядку, і так до кінця, після чого не залишиться один порожній масив, який ми теж видаляємо.

6. Розмір статичного масиву задається один раз, та не змінюється в ході алгоритму, коли динамічний масив змінює свій розмір.

8. Ми заповнюємо масив `mas[]`; за допомогою покажчика `*pointMas`, що збільшується із одночасно з змінною `i`, та записує її значення до масиву `mas`.

9. Для того, щоб не виник виток пам'яті.

Висновок: Я виконала лабораторну роботу 11, використовуючи вказівники. Створила програму, яка визначає кількість парних додатних чисел у кожному рядку.