Лабораторна робота №7. Функції

**Автор:** Панкеєв Владислав

**Група:** КН-922Б

**Завдання:**

**1.**Переробити програми, що були розроблені під час виконання лабораторних робіт з тем "Масиви" та "Цикли" таким чином, щоб використовувалися функції для обчислення результату.

**2.**Функції повинні задовольняти основну їх причетність - уникати дублювання коду.

Тому, для демонстрації роботи, ваша програма (функція main()) повинна мати можливість викликати розроблену функцію з різними вхідними даними.

**3.**Слід звернути увагу: параметри одного з викликів функції повинні бути згенеровані за допомогою генератора псевдовипадкових чисел random().

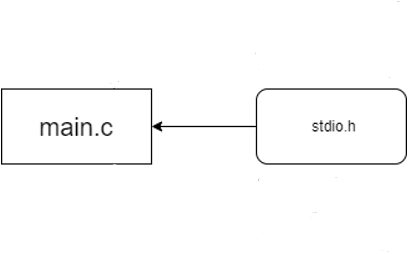
**Опис програми**

***Функціональне призначення***

Ця програма виконує дві операції.

* Ділить символьний масив на 2 частини та міняє їх місцями
* Визначає кількість парних чисел у заданому діапазоні

***Опис логічної структури***

**

*(Рис. 1) Графічна структура програми*

***Файл "main.c"***

Головний файл

Це файл, який містить точку входу main, функції lab5 та lab6 .

int main()

{

lab5();

lab6();

}

## Головна функція.

Містить у собі виклик другорядних функцій lab5 та lab6, які містять код програми для розрахунку множення матриці саму на себе та перевірку на просте чи не просте число.

**Послідовність дій**

* Присвоїти значення аргументам a і b, значення цих аргументів ми передаємо у функцію lab5 .
* Виклик функції lab5, у параметрах цієї функції при виклику вказуємо аргументи a і b.
* Викликаємо функцію iterator, у параметрах цієї функції при виклику нічого не ваказуємо.

*int a, b, k = 0;*

*printf("Введіть початок діапозону:");*

*scanf("%d", &a);*

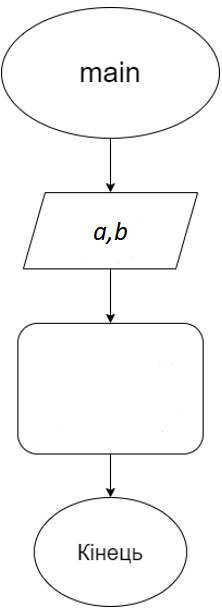
*printf("Введіть кінець діапозону:");*

*scanf("%d", &b);*

*for (int iterator = a + 1; iterator < b; iterator ++){ //Продувжуємо операцію до тих пір, поки Ітератор не стане більше b*

*if (iterator % 2 == 0){ //Якщо Ітератор при діленя на 2 має остачу 0, додаємо до k 1.*

*k++;*

**

*(Рис. 2) Схема алгоритму функції main*

void lab5 ()

Ця визначає парну кількість елементів у діапазоні

**Аргументи**

a – початок діапазону.

b - кінець даіпазону.

**Послідовність дій**

* Вводимо початок на кінець діапазону
* Перевіряємо чи наша цифра не вийшла за кінець нашого діапазону
* Якщо перевірка була додаємо до k 1 і продовжуємо операцію

for (int iterator = a + 1; iterator < b; iterator ++){ //Продувжуємо операцію до тих пір, поки Ітератор не стане більше b

if (iterator % 2 == 0){ //Якщо Ітератор при діленя на 2 має остачу 0, додаємо до k 1.

k++;

}

}

void lab6 ()

Функція ділить символьний масив на 2 частини та міняє їх місцями

**Послідовність дій**

* Вводимо символьний масив.
  + Потім рахуємо його розмір та обов’язково віднімаємо ітеруючий нуль*.*

char words[] = "lvanov";

int size = sizeof(words)-1;

printf("Array size: %i\n", size);

* Потім за рахунок тимчасових змінних заміняємо символи

char temp = words[i];

words[i] = words[size/2+i];

words[size/2+i] = temp;

***Структура проекту лабораторної роботи:***

├── lab07

├── Makefile

├── README.md

└── src

└── main.c

**Висновки:** У цій роботі було перетворено лабораторні проекти №5 та №6 для використання функцій. Було набуто навичок роботи з функціями, їх декларація, реалізація та виклик. Під час тестування програми були отримані результати функції lab5 – визначення кількості парних символів у діапазоні, та функції lab6 – розділення символьного масиву на 2 частини та зміни їх місцями.