**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 5

**На тему:** *“основи роботи з функціями в C ”*

**З дисципліни:** *“Основи програмування”*

**Лектор:**

ст.викл. каф. ПЗ

Муха Т.О.

**Виконав:**

ст. гр. ПЗ-15

Марущак А.С.

**Прийняла:**

асист. каф. ПЗ

Заводовська Н.О.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2021 р.

∑= \_\_\_\_ .

Львів – 2021

**Тема роботи:** основи роботи з функціями в С.

**Мета роботи:** здобути практичні навики створення тазастосування функцій у мові С .

**Теоретичні відомості**

У процесі програмування реальних практичних задач дуже часто виникає ситуація коли на різних етапах розв’язання вихідної задачі доводиться неодноразово розв’язувати деяку часткову підзадачу. Причому алгоритм розв’язання цієї підзадачі передбачає виконання тої ж самої послідовності дій, але над різними наборами даних. Для підвищення ефективності програми та її простоти і наглядності, групу операторів, які реалізують таку послідовність дій, оформляють у вигляді самостійної програмної одиниці (підпрограми). У мові С такі підпрограми реалізуються у вигляді функцій. Вони записуються в коді програми лише один раз, а у відповідних місцях програми (тобто там де потрібно розв’язати цю часткову підзадачу) забезпечується лише звертання до них (виклик функції). Така техніка розробки програм, поперше, дозволяє уникнути повторення в програмі окремих фрагментів коду, а, по-друге, забезпечує повторне використання програмного коду, тобто, використання існуючих функцій як стандартних блоків для створення нових програм. Для цього кожна функція повинна розв’язувати одну чітко визначену задачу, а ім’я функції має наочно відображати цю задачу. Тоді сама програма на С, а точніше головна функція main(), буде просто містити виклики таких функцій у потрібному порядку, який забезпечує виконання алгоритму розв’язання вихідної задачі.

Синтаксис мови С передбачає такий формат опису функції:

[тип\_результату] ім’я\_функції(список\_параметрів){тіло функції;}

Поле тип\_результату задає тип значення, що повертається функцією. У ролі типу результату функції може використовуватися ключове слово void, яке означає, що функція не повертає ніякого значення. Поле ім’я\_функції задає унікальний в межах програми ідентифікатор за допомогою якого можна звертатися до функції (викликати функцію). В мові С він трактується як особливий тип вказівника, який називається вказівником на функцію. Значенням цього вказівника є адреса точки входу в функцію. Зустрівши визначення функції, компілятор створює самостійну секцію коду програми, що на етапі компонування об'єднується з іншими функціями. Початкова адреса цієї секції і є значенням вказівника на функцію. Поле список\_параметрів представляє собою список так званих параметрів для позначення вхідних даних функції. Тому ці параметри часто називають формальними параметрами. З точки зору синтаксису мови С кожний формальний параметр описується аналогічно до звичайної змінної, тобто опис формального параметру включає задання типу

даних та ідентифікатора. Якщо функція потребує декілька формальних параметрів, то вони задаються у списку послідовно, розділяючись комою, але кожний параметр у цьому списку описується за наведеним вище правилом (тобто для кожного параметру треба вказати свій тип, навіть якщо всі вони мають одинаковий тип). У випадку, коли функція не має параметрів, список формальних параметрів або задається ключовим словом void, або є порожнім (але круглі дужки у цьому випадку залишаються). Сукупність різних оголошень та операторів, які знаходяться між парою фігурних дужок в описі функції утворюють тіло функції.

Існують два способи завершення виконання функції і повернення у точку програми, з якої здійснено виклик. Перший - завершення виконання функції з використанням оператора повернення return. Інший спосіб завершення виконання функції полягає у послідовному проходженні всіх операторів тіла функції до закриваючої фігурної дужки. Цей спосіб можливий лише для функцій з типом результату void, хоча як показує наступний приклад він не є єдино можливим способом виходу з функцій типу void.

Виклик функції здійснюється в потрібному місці програми за її іменем (аналогічно як і для стандартних бібліотечних функцій) у такому загальному форматі:

ім’я\_функції([список\_аргументів])

Поле список\_аргументів задає значення фактичних параметрів, які підставлються у відповідні формальні параметри. На відміну від списку формальних параметрів у списку аргументів не потрібно вказувати тип аргумента, достатньо записати лише ідентифікатор. В загальному випадку аргумент може задаватися виразом відповідного типу, тобто типу який співпадає з типом відповідного формального параметра в описі функції. Під час виклику функції у формальні параметри підставляються значення аргументів у тому порядку, в якому вони задані при виклику функції. Кількість формальних параметрів функції та аргументів повинні співпадати (за винятком випадку коли формальні параметри мають значення за замовчуванням). Якщо у функції відсутні формальні параметри, то при виклику такої функції повинен також бути відсутнім і список аргументів (але пара дужок () при виклику залишається).

**ЛАБОРАТОРНЕ ЗАВДАННЯ**

1. Ознайомитися з теоретичним матеріалом викладеним вище в даній інструкції і виконати приклади програм.

2. Одержати індивідуальне завдання.

3. Розробити алгоритм розв’язання індивідуального завдання і подати його у вигляді блок-схеми.

4. Скласти програму на мові С у відповідності з розробленим алгоритмом.

5. Виконати обчислення по програмі.

6. Підготувати та здати звіт про виконання лабораторної роботи.

**Індивідуальне завдання**

Протабулювати, задану згідно варіанту функцію, на проміжку [a,b] з кроком h двома способами. Для обчислення значення, заданої згідно варіанту функції y в точці x∈[a,b]:

1) першим способом: оголосити і реалізувати функцію мовою С, формальним параметром якої є x, а результатом функції є значення y;

2) другим способом: оголосити і реалізувати функцію мовою С, першим формальними параметром функції є x, другим формальним параметром є аргумент, куди буде повернено результат обчислення за формулою.

Функція повертає:

Табуляцію оформити у вигляді окремої функції. Результати обчислень подати у вигляді таблиці. Всі функції розмістити в заголовному файлі.

23. f=(3x2+x−2)/(3x+1),a=0,b=1;

Код програми :

lab05.c

#include <stdio.h>

#include <conio.h>

#include "lab05\_func.h"

int main(){

    double a = 0, b = 0, h = 0;

    printf("Enter a: ");

    scanf("%lf", &a);

    printf("Enter b: ");

    scanf("%lf", &b);

    printf("Enter step: ");

    scanf("%lf", &h);

    tabulate(a, b, h);

    tabulate2(a, b, h);

    getch();

}

lab05\_func.h

#ifndef MY\_FUNCTIONS

#define MY\_FUNCTIONS

double func(double x);

int func2(double x, double\* y);

void tabulate(double a, double b, double h);

void tabulate2(double a, double b, double h);

#endif

lab05\_func.c

#include <stdio.h>

double func(double x){

    return (3\*x\*x + x - 2)/(3\*x + 1);

}

int func2(double x, double \*y){

    double res = func(x);

    \*y = res;

    if(res < 1e-6 && res > -1e-6){

        return 0;

    }

    else if (res > 0)

    {

        return 1;

    }

    else{

        return -1;

    }

}

void tabulate(double a, double b, double h){

    if(a < b && h < b-a){

        printf("1)\n");

        printf("%-10s | %-10s\n", "X", "Y");

        for(double x = a; x <= b; x += h){

            printf("------------------------\n");

            printf("% -10.4lf | % -10.4lf\n", x, func(x));

        }

    }

    else{

        printf("Incorrect parameters!");

    }

}

void tabulate2(double a, double b, double h){

    if(a < b && h < b-a){

        double y = 0;

        int res = 0;

        printf("2)\n");

        printf("%-10s | %-10s | %-10s\n", "X", "Y", "Return");

        for(double x = a; x <= b; x += h){

            printf("------------------------------------\n");

            res = func2(x, &y);

            printf("% -10.4lf | % -10.4lf | % -10d\n", x, y, res);

        }

    }

    else{

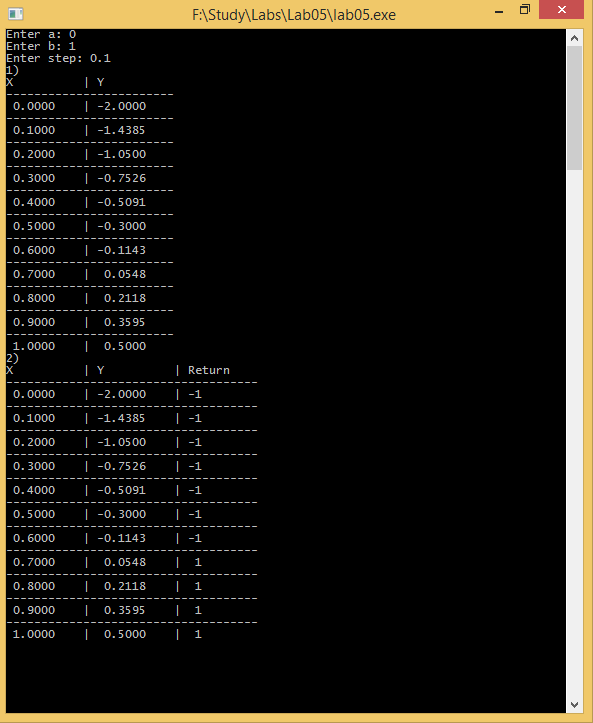
        printf("Incorrect parameters!");

    }

}

**Протокол роботи:**

Програма отримує крок та межі табулювання і виводить таблиці за допомогою функцій:



Програма виводить 2 таблиці для двох варіантів функцій.

**Висновок:**

Ми написали програму, яка табулює функцію, реалізовану на мові С за допомогою функцій табуляції.