**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **ІКНІ**

Кафедра **ПЗ**

**ЗВІТ**

До лабораторної роботи № 2

**На тему:** *“Програмування Циклічних Процесів в С”*

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

**Лектор:**

ст.викл. каф. ПЗ

Муха Т. О.

**Виконав:**

гр. ПЗ - 18

Таранець Є. А.

**Прийняв:**

ст.викл. каф. ПЗ

Кутельмах Р.К.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_ .

Львів – 2022

**Тема роботи:** програмування циклічних процесів в С

**Мета роботи:** навчитися організовувати програми циклічної структури, які дозволяють  
повторювати певну групу операторів задану кількість разів.

**Лабораторне завдання**

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

приклади програм.

2. Одержати індивідуальне завдання з Додатку 1.

3. Розробити алгоритм розв’язання індивідуального завдання і подати його у вигляді блоксхеми.

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

5. Виконати обчислення по програмі.

6. Одержати індивідуальне завдання з Додатку 2.

7. Розробити алгоритм розв’язання індивідуального завдання і подати його у вигляді блоксхеми.

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

9. Виконати обчислення по програмі при різних значеннях точності і порівняти отримані

результати.

10.Підготувати та здати звіт про виконання лабораторної роботи

**Індивідуальне завдання**

**Завдання 1:** Використовуючи цикли знайти і вивести всі чотиризначні натуральні числа, сума цифр яких не перевищує 23.

**Завдання 2:** З допомогою операторів циклу, протабулювати на відрізку від А до В з області визначення функцію, задану розкладом у ряд Тейлора. Для порівняння обчислити також у кожній точці табуляції значення функції задане формулою. Результати подати у виді таблиці з коментарями.

**Теоретичні відомості**

**Поняття про цикли**

Алгоритм називається *циклічним*, якщо певна послідовність однотипних дій (тіло циклу)  
виконується багато разів. Однократне виконання тіла циклу називається *ітерацією*. В  
залежності від постановки задачі розрізняють *арифметичні* цикли (тобто цикли з наперед  
відомою кількістю ітерацій) та *ітераційні* цикли (число ітерацій такого циклу наперед  
невідомо). Отже, цикли дозволяють багатократно виконувати сукупність однотипних операцій. Це означає, що програмний код за рахунок циклів виглядатиме компактнішим. Умовою виходу  
циклу або ж умовою продовження циклу є вираз, від істинності якого залежить, буде  
виконуватися тіло циклу чи ні. **while**, **do…while** та **for**. Усі три оператори циклу в С  
можуть використовуватися для реалізації як арифметичних, так й ітераційних циклічних  
процесів. Цикл while є циклом *з передумовою*, а цикл do…while – *з післяумовою*. В циклах з  
передумовою *спочатку* перевіряється умова, а *тоді*, залежно від того, істинна вона чи хибна,  
виконується або не виконується тіло циклу. У циклах з післяумовою *спочатку* виконується тіло  
циклу, а *тоді* перевіряється умова, від істинності якого залежить виконання *наступної* ітерації  
циклу. Одне з типових застосувань циклу while – перевірка, чи не досягнуто кінець файлу. Цикл  
for, як правило, застосовується при заздалегідь відомій кількості ітерацій – наприклад, для  
опрацювання всіх елементів масиву.

**Конструкція for**

Цикл **for** управляється змінною, що називається лічильником циклу. Величину зміни  
лічильника називають кроком. Синтаксис циклу є наступний:

**for(<початкова\_інструкція>; <умова>; <вираз>)**

**<тіло циклу>;**

Спочатку виконується початкова інструкція та перевіряється умова. Якщо вона істинна, то  
виконуються інструкції з тіла циклу, а тоді обчислюється вираз та управління передається в  
початок циклу з тією різницею, що початкова інструкція вже не виконується.

**Конструкція while**

Цикл – while використовують в тому випадку, коли необхідно, щоб на початку перевірялася деяка умова, а потім, якщо умова істинна, виконувалися деякі дії. В процесі виконання цих дій відбувається зміна значень змінних, що входять до вказаної умови у якості операндів. Завдяки цьому при N-кратному виконанні циклу значення логічного виразу – умови з True на False.

**while** (<**умова**>)

**<тіло циклу>;**

**Хід роботи**

**1.Блок-схема до першого завдання:**

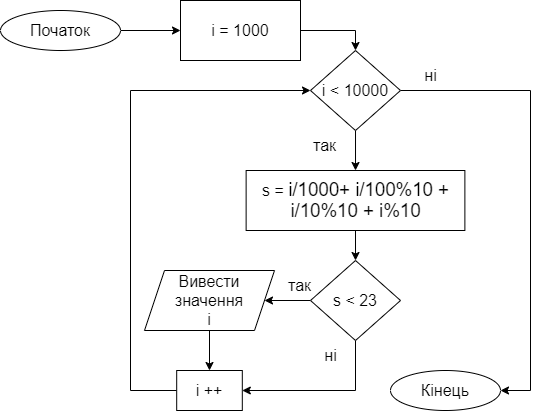
****

Рисунок 1. Блок-схема першого завдання

**Код програми:**

**Назва файлу: Lab2\_1.c**

#include <stdio.h>  
#include <conio.h>  
int main()  
{  
for(int i=1000;i<10000;i++) {  
int s=i/1000+ i/100%10 + i/10%10 + i%10;  
 if (s <= 23) {  
 printf("%d\t", i);  
 }  
}  
 getch();  
return 0;  
}

**Протокол роботи:**

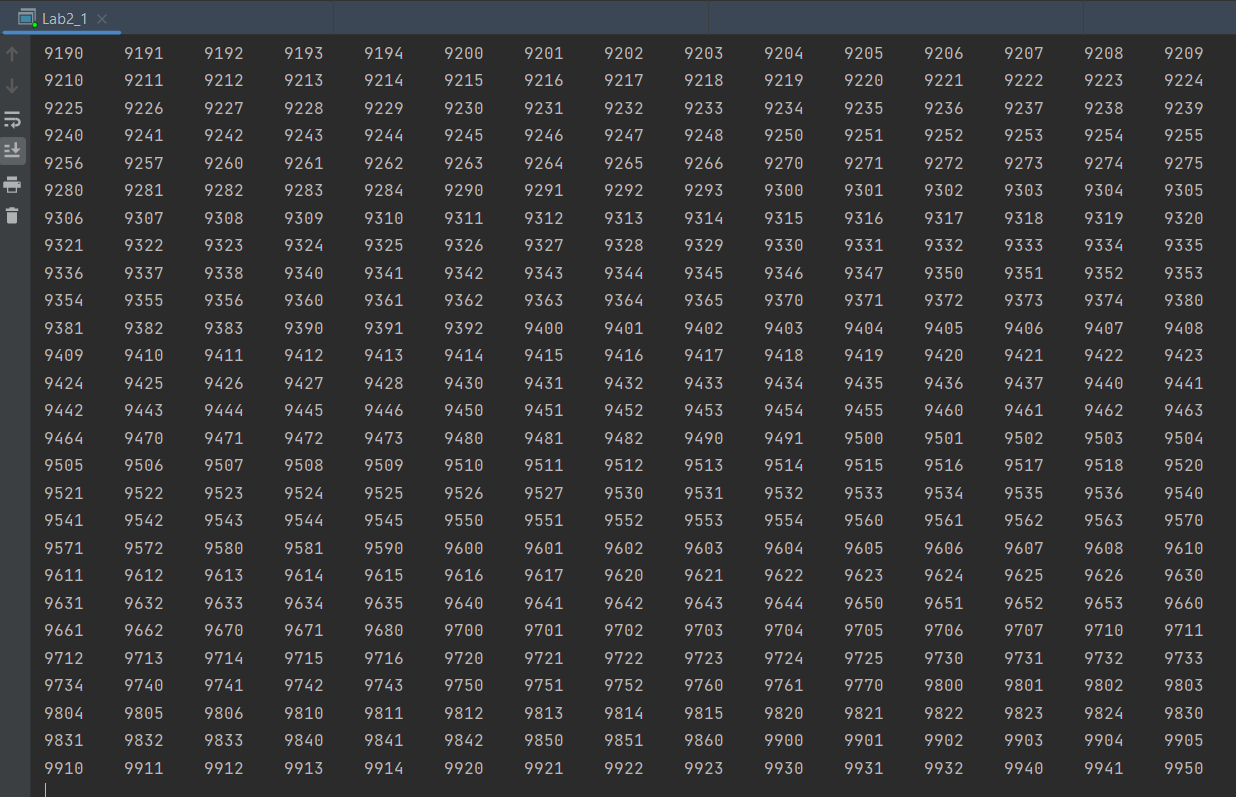


Рисунок 2. Приклад виведення розв'язку першого завдання

**Висновок:** розроблено алгоритм виводу усіх чотирьохзначних чисел,

сумма яких < 23, використовуючи оператор циклу for, та умовний оператор if.

**2. Блок-схема до другого завдання:**

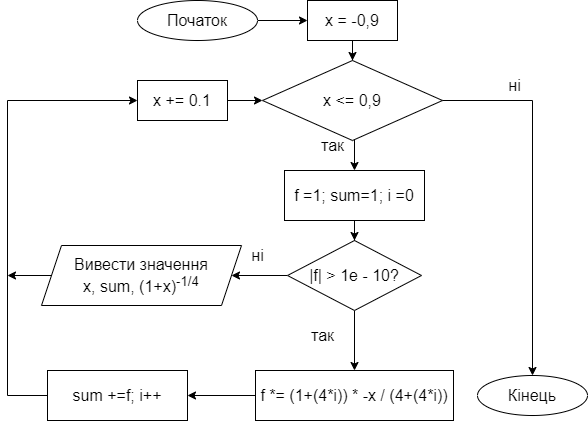
****

Рисунок 3. Блок-схема другого завдання

**Код програми:**

**Назва файлу: Lab2\_2.c**

#include <stdio.h>  
#include <math.h>  
int main() {  
 double f,sum,x;  
 printf("-------------------------------------------------\n");  
 printf("|\t%s\t| %9s\t| %11s\t|\n", "X","Taylor", "Function");  
 printf("-------------------------------------------------\n");  
 for(x=-0.9;x<=0.9;x+=0.1) {  
 f=1;sum = 1;int i=0;  
 while(fabs(f)>1e-10)  
 {  
 f= f \* (1+(4\*i))\* -x/(4+(4\*i));  
 sum += f;

i++;  
 }  
 printf("|%10.1f\t|%12.6f\t|%12.6f\t|\n",x, sum,pow(1.0+x,-1.0/4));  
 }  
 printf("-------------------------------------------------\n");  
 return 0;  
}

**Протокол роботи:**

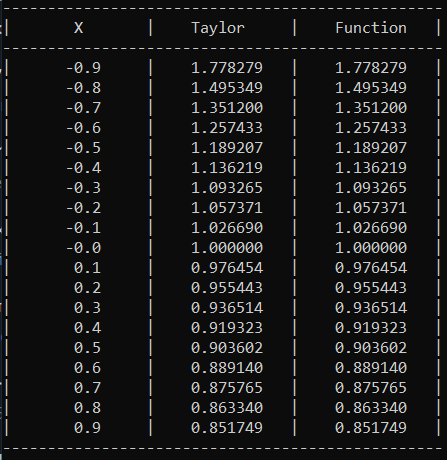


Рисунок 4. Приклад виведення розв'язку другого завдання

**Висновок:** розроблено алгоритм для обчислення ряду Тейлора, використовуючи 2 оператора циклу (for і while). Виведення обчислених значень за допомогою ряду Тейлора і значень обчислених функцією, для порівняння у вигляді таблиці.