Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Алгоритмізація та програмування

Базові концепції програмування

ЗВІТ ДО

ЛАБОРАТОРНОЇ РОБОТИ № 3

Варіант №17

Виконав: студент 1-го курсу

гр. ТР-12

Каркушевський В.Л. (П.І.Б.)

Оцінка \_\_\_\_\_\_\_\_\_ Перевірила: Смаковська Г.М

Дата «14»Листопада 2021 \_\_\_\_\_\_\_\_\_\_\_(П.І.Б., підпис)

Київ – 2021

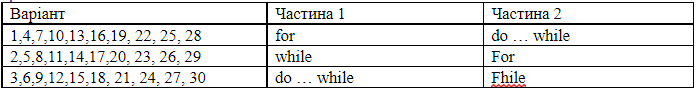
**Частина 1**

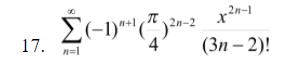
Обчислити значення функції **y=f(x)** та видати результат у вигляді таблиці, вважаючи, що аргумент **x** змінюється на довільно заданому інтервалі з довільно заданим кроком. Функцію взяти з індивідуального завдання до л/р №1. Для **парних** варіантів задається **величина кроку**, для **непарних** – кількість значень на інтервалі.

**Частина 2**

Обчислити суму членів ряду з точністю до члена ряду, меншого еps. Вивести значення кожного члена ряду.

***Завдання на лабораторну роботу:***





**Теоритичні відомості**

**Циклом** називається блок коду, який для вирішення завдання потрібно повторити кілька разів.

Кожен **цикл** складається з:

* + блоку перевірки умови повторення циклу
  + тіла циклу

Цикл виконується доти, доки блок перевірки умови повертає справжнє значення.

Тіло циклу містить послідовність операцій, що виконується у разі істинної умови повторення циклу. Після виконання останньої операції тіла циклу знову виконується операція перевірки умови повторення циклу. Якщо ця умова не виконується, то буде виконано операцію, що стоїть безпосередньо після циклу в коді програми.

**У мові Сі такі види циклів:**

* while - цикл з передумовою;
* do ... while - цикл з постумовою;
* for — параметричний цикл (цикл із заданим числом повторень).

Цикл з передумовою **while**

Якщо Умова виконується (вираз, що перевіряє умову, не дорівнює нулю), то виконується блок операцій, укладений у фігурні дужки, потім умову перевіряється знову.

Послідовність дій, що складається з перевірки умови та виконання блоку операцій, повторюється доти, доки вираз, що перевіряє умову, не стане хибним (рівним нулю). При цьому відбувається вихід із циклу, і виконується виконання операції, що стоїть після оператора циклу.

Цикл з післяумовою **do...while**

це цикл із післяумовою, де істинність виразу, що перевіряє умову перевіряється після виконання блоку операцій, укладеного у фігурні дужки. тіло циклу виконується до тих пір, поки вираз, що перевіряє умову, не стане хибним, тобто тіло циклу з постумовою виконається хоча б один раз.

Використовувати цикл do...while краще у тих випадках, коли має бути виконана хоча б одна ітерація, або коли ініціалізація об'єктів, що беруть участь у перевірці умови, відбувається всередині тіла циклу.

Параметричний цикл **for**

Параметричний цикл (цикл із фіксованою кількістю повторень). Для організації такого циклу необхідно здійснити три операції:

* Ініціалізація - надання параметру циклу початкового значення;
* Умова – перевірка умови повторення циклу, найчастіше – порівняння величини параметра з деяким граничним значенням;
* Модифікація - зміна значення параметра наступного проходження тіла циклу.

Ці три операції записуються в дужках і поділяються крапкою з комою;;. Як правило, параметром циклу є ціла змінна. Ініціалізація параметра здійснюється лише один раз – коли цикл for починає виконуватися.

**Опис завдання:**

Оголошуємо директиви : #include <stdio.h> , #include <math.h**>,** та математичну константу #define PI\_4 0.785398163397448309616

Створюємо три методи: task1(виконання 1 частини), task2(виконання 2 частини та dfactorial(для обчислення факторіала).

В методі task1 за допомогою команди scanf вводимо значення змінних з клавіатури(a,b,x-початкова границя інтервала, krok-крок функції, n - кількість значень на інтервалі)

За допомогою оператора while обраховуємо функцію.

В методі task2 за допомогою команди scanf вводимо значення змінних з клавіатури(eps,x)

За допомогою оператора for обчислюємо суму членів ряду з точністю до члена ряду, меншого за еps. Виводимо значення кожного члена ряду.

В main викликаємо метод task1 та task2 за допомогою функції printf.

**Програмний код**

#include <stdio.h>

#include <math.h>

#define PI\_4 0.785398163397448309616

float dfactorial(float danswer);

int zavd1();

int zavd2();

int main(void) {

printf("\t\t\tЗавдання 1\n");

printf("%d", zavd1());

printf("\n\t\t\tЗавдання 2\n");

printf("%d", zavd2());

}

int zavd1(){

float a,b,x,krok ;

float y ;

printf("Введіть значення для обрахунку функції\n");

printf("a = ");

scanf("%f", &a);

printf("b = ");

scanf("%f", &b);

printf("Введіть початкову границю інтервала\n");

printf("x = ");

scanf("%f", &x);

printf("Введіть крок\n");

printf("krok = ");

scanf("%f", &krok);

float n ;

printf("Введіть кількість значень на інтервалі \nn = ");

scanf("%f", &n);

float k = x + n \* krok;

printf("Крок : %.4f, початкова границя інтервала : %.4f, кінцева границя інтервала: %.4f\n",krok , x, k);

while(x < k ){

y=2. / 7 \* b \* powf( x, 1. / 3) + exp(fabs(x)) + log(a \* b \* x );

x += krok;

printf("f(x) = %.4f \n", y);

}

printf("Кінець обрахунків\n");

return (0);

}  
int zavd2(){

double eps , s = 0 , x , y;

int n = 1;

printf("\n Введіть значення eps: ");

scanf("%lf", &eps);

printf("\n Введіть значення x :");

scanf("%lf", &x);

printf("\n");

y = pow(-1 , n + 1) \* pow( PI\_4 , (2 \* n) - 2 ) \* pow(x , 2 \* n - 1 ) / (dfactorial((3\*n)-2)) ;

for( int n =1;y > eps ;n++){

y = pow(-1 , n + 1) \* pow( PI\_4 , (2 \* n) - 2 ) \* pow(x , 2 \* n - 1 ) / (dfactorial((3\*n)-2)) ;

s+= y ;

printf("y = %lf\n", y );

printf("Сума при n = %d : %lf\n\n",n,s);

}

printf("Кінець обрахунків\n");

return 0;

}

float dfactorial(float danswer)

{

if(danswer <= 1.0)

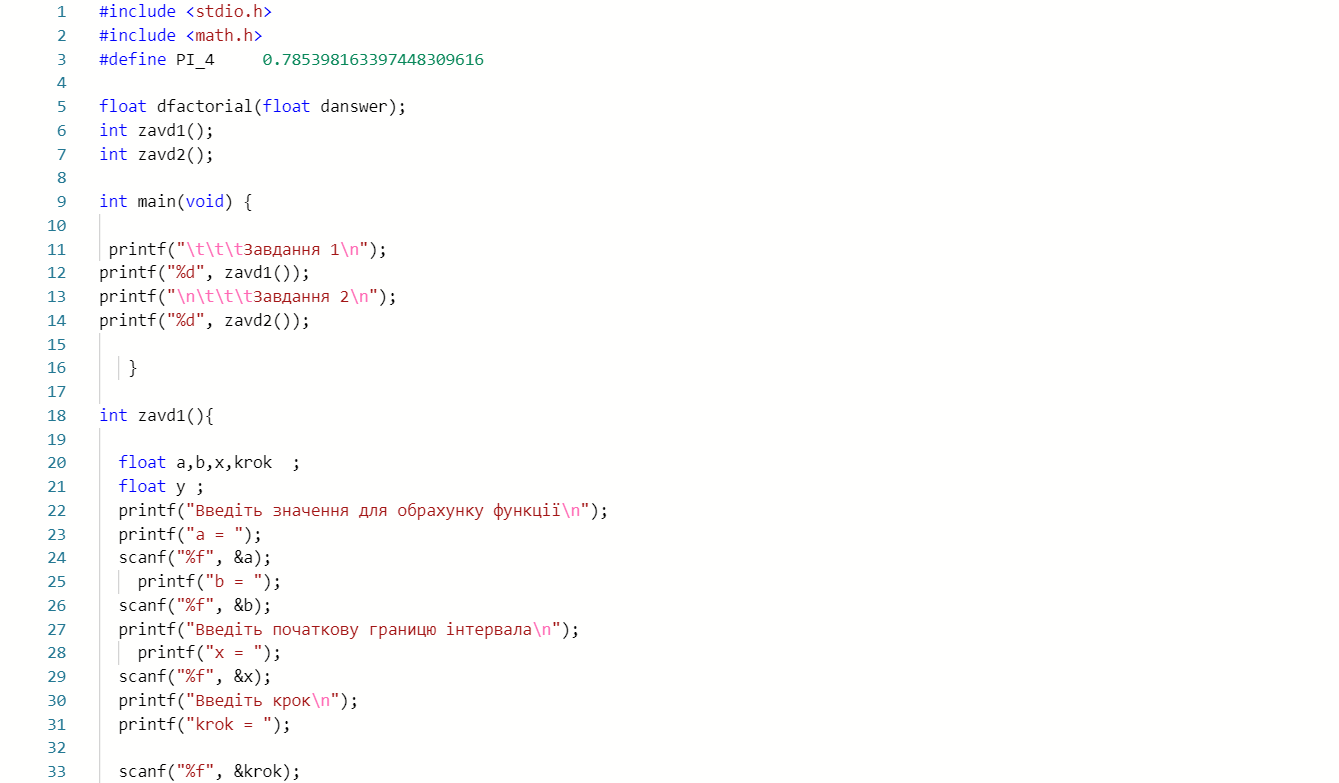
return(1.0);

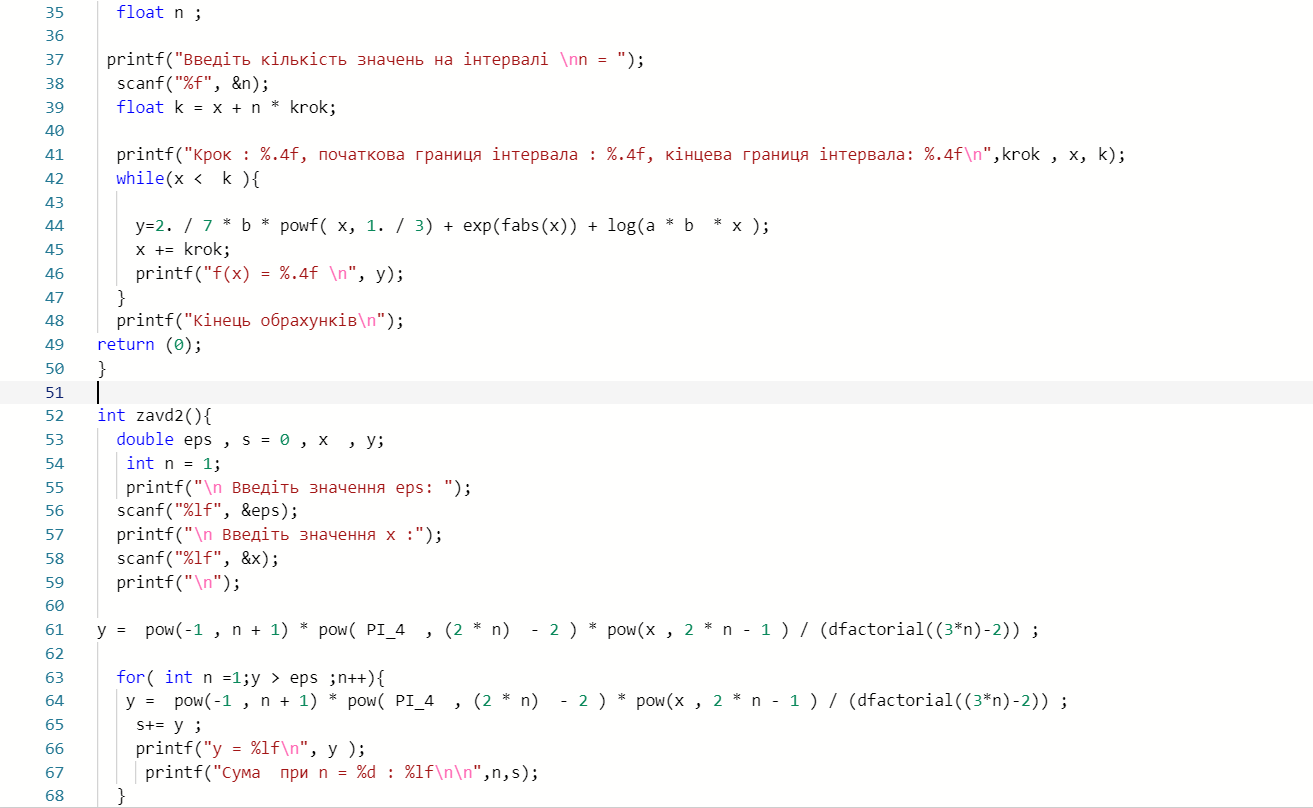
else

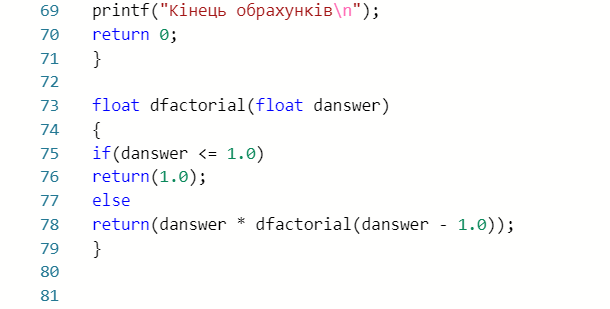
return(danswer \* dfactorial(danswer - 1.0));

}

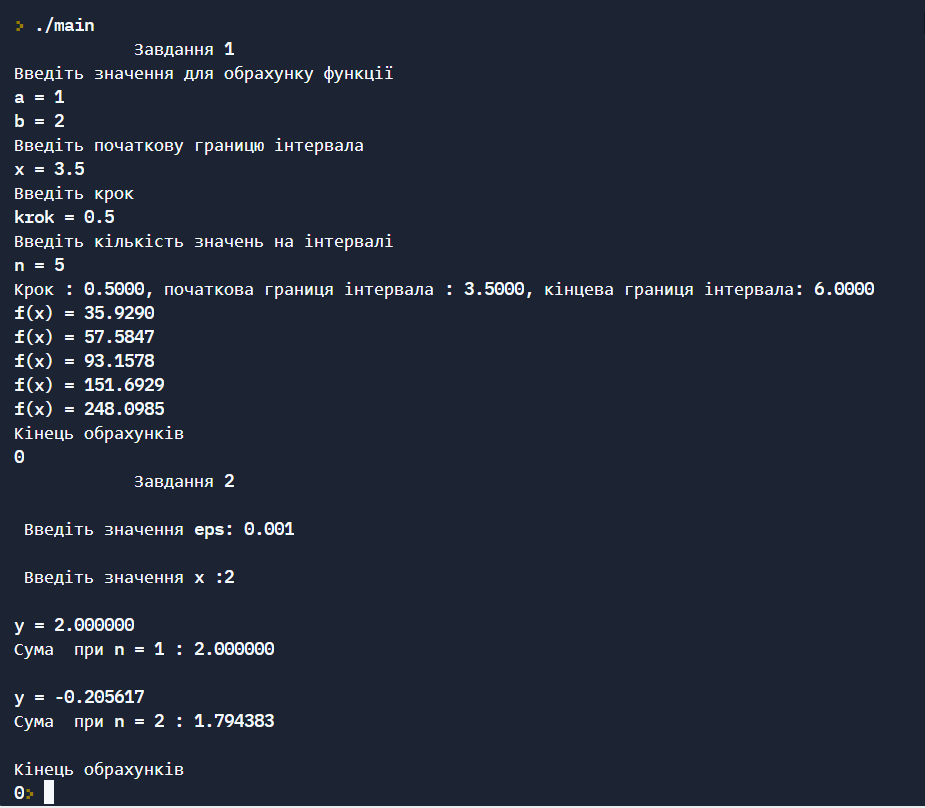
**Код програми:**







**Результати рішення програми :**



**Висновок:**

При виконанні лабораторної роботи було досліджено використання операторів for та while. Виконано математичні розрахунки за заданим завданням враховуючи особливостей.Також попрацював з факторіалом.