МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Програмування. Частина 1. Основи програмування

Лабораторна робота №5

Тема: Реалізація алгоритмів з використанням функцій та вказівників

Варіант № 10

Виконав:

студент групи ІС-34

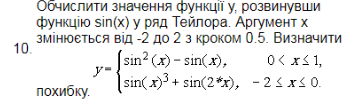
Колосов Ігор

Перевірив:

Драган М.С.

Київ-2023

**Завдання №1**



Вирішення:

#include <iostream>

#include <cmath>

typedef double ( \*sinT )(double x);

double sinTaylor(double x) {

double sum = 0, tempValue = 1;

for (int i = 0; fabs(tempValue) > 0.001; i++, sum+=tempValue)

tempValue = pow(-1, i) \* pow(x, 2 \* i + 1) / tgamma(2\*i+1);

return sum;

}

double function(sinT sinus, double x) {

double y = -1;

if (0 < x && x <= 1) y = pow(sinus(x), 2) - sinus(x);

else if (-2 < x && x <= 0) y = pow(sinus(x), 3) - sinus(2\*x);

return y;

}

int main() {

double \*res = new double;

const float LOWER\_BOUND = -2, UPPER\_BOUND = 2, STEP = 0.5;

for (double i = LOWER\_BOUND; i <= UPPER\_BOUND; i += STEP) {

*// ВИКОРИСТАНО ВКАЗІВНИКИ НА ФУНКЦІЇ*

double fTailor = function(sinTaylor, i);

double fOrigin = function(sin, i);

std::cout

<< " <iteration: " << UPPER\_BOUND - LOWER\_BOUND / STEP

<< "; x: " << i << ">\n"

<< " > fTailor: " << fTailor << "\n"

<< " > fOrigin: " << fOrigin << "\n"

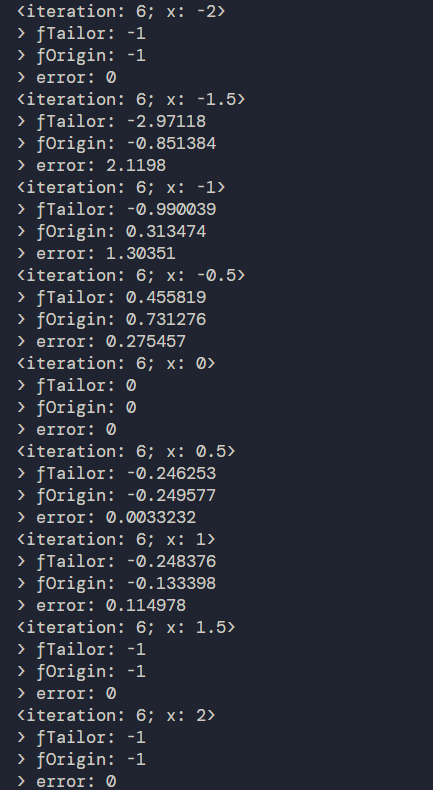
<< " > error: " << fabs(fTailor - fOrigin) << "\n";

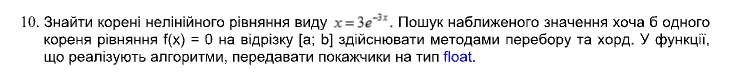
}

return 0;

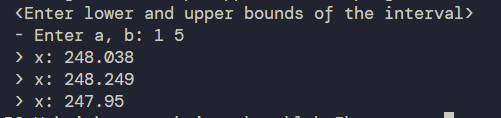
}

Вивід:



**Завдання №2**

Вивід:



#include <iostream>

#include <math.h>

typedef double ( \*func )(double x);

double chord(func f, double x0, double x1) {

double x2 = x1 - (x1 - x0) \* (f(x1) - f(2 \* x1)) / (f(x1) - f(2 \* x1) - f(x0) + f(2 \* x0));

if (fabs(x2 - x1) < 0.001) return x2;

return chord(f, x1, x2);

}

double chord\_p(func f, double \*x0, double \*x1) {

double x2 = \*x1 - (\*x1 - \*x0) \* (f(\*x1) - f(2 \* (\*x1))) / (f(\*x1) - f(2 \* (\*x1)) - f(\*x0) + f(2 \* \*x0));

if (fabs(x2 - \*x1) < 0.001) return x2;

return chord\_p(f, x1, &x2);

}

double function(double x) {

return 3 \* pow(exp(1.0), -3\*x);

}

int main() {

double limA, limB;

std::cout << " <Enter lower and upper bounds of the interval>\n" << " - Enter a, b: ";

std::cin >> limA >> limB;

for (double i = limA; i < limB; i += 0.1) {

double j = i + 0.1;

double x = chord\_p(function, &i, &j);

std::cout << " > x: " << x << "\n";

}

return 0;

}

**Контрольні запитання:**

1. Дати поняття посилальних типів даних?
   1. Тип значення якого задає посилання
2. Як здійснюється доступ до значення змінної?
   1. Розіменування вказівника
3. Як оголосити вказівник на певний тип та посилання на змінну?
   1. Тип\_даниих \*вказівник = &посилання
4. Як ініціалізувати вказівник та посилання?
   1. Тип\_даних \*вказівник
   2. Тип\_даних &посилання = на що посилання
5. Що відбувається під час звернення до неініціалізованого вказівника?
   1. Ми отримаєм «сміття», тобто рандомне значення з купи пам’яті
6. Які значення можна присвоювати вказівнику?
   1. Посилання
7. Які операції припустимі для вказівника?
   1. Присвоєння, порівняння та розіменування
8. Що таке «розіменування вказівника»?
   1. Отримання значення з вказівника
9. Як змінюється значення вказівника при додаванні чи відніманні цілого числа?
   1. У нас переводиться ціле число в байти і змінюється посилання вказівника, наприклад як в масиві номер елементу
10. У чому полягає особливість вказівників типу void\*?
    1. Вказівник типу void може приймати будь яке значення
11. Як використовують вказівники на функції?
    1. Для вказування на показник адресу функції, або ж вивід функції

**Висновок:**

Ознайомився з особливостями посилальних типів даних, опанував технологію застосування цих же типів даних та навчився розробляти алгоритми та програми із застосуванням посилальних типів даних.