Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

Звіт

з лабораторної роботи № 3 з дисципліни «Основи програмування-1»

«Організація циклічних процесів. Ітераційні цикли»

Варіант 29

Виконав студент ІП-02 Глазунова Поліна Андріївна

(шифр, прізвище, ім'я, по батькові)

Перевірив Вітковська Ірина Іванівна

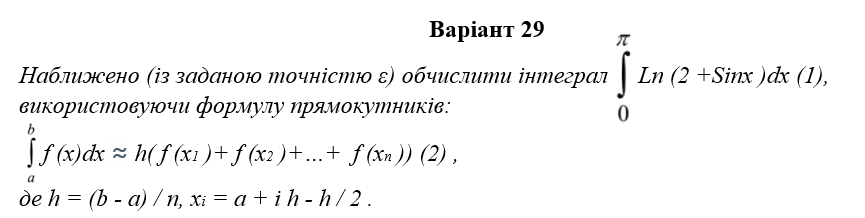
( прізвище, ім'я, по батькові)

Київ 2020

**Лабораторна робота 3**

**Організація циклічних процесів. Ітераційні цикли**

**Мета –** вивчити особливості організації ітераційних циклів.



***Python:***

from math import sin

import math

e = float(input("Enter the accuracy: ")) # Стала ε (точність обчислювання)

n = 1 # Кількість відрізків n, на яку розбили відрізок інтегрування

h = math.pi / n # Висота кожного окремого прямокутника

s = math.log(2 + math.sin(math.pi / 2 \* n)) \* h # Поточне наближення інтегралу

while True:

s0 = s # Попереднє наближення інтегралу

n = 2 \* n

s = 0.0

h = math.pi / n

for i in range (n):

x = math.log(2 + math.sin(i \* math.pi / n - math.pi / 2 \* n )) \* h

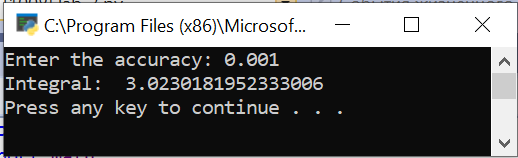
s = s + x

if math.fabs(s - s0) < e:

break

Res = s # Шукане наближене значення інтеграла

print('Integral: ', Res)



***C++:***

#include <iostream>

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include <cmath>

using namespace std;

int main() {

double e;

cout << "Enter the calculation accuracy = ";

cin >> e; //введення точності обчислень

double h; //висота кожного прямокутника

double s; //поточне наближення інтегралу

double Res; //кінцевий результат (наближене значення інтегралу)

double s0; //попереднє наближення інтегралу

int n = 1; //кількість відрізків, на яку ми розбиваємо відрізок інтегрування

h = M\_PI / n;

s = h \* log(2 + sin(M\_PI / 2 \* n));

do {

s0 = s; //попереднє наближення інтегралу

n = 2 \* n;

s = 0.0;

h = M\_PI / n;

for (int i = 1; i <= n; i++)

{

double x = h \* log(2 + sin(i \* M\_PI / n - M\_PI / 2 \* n)); //аргумент х

s += x;

}

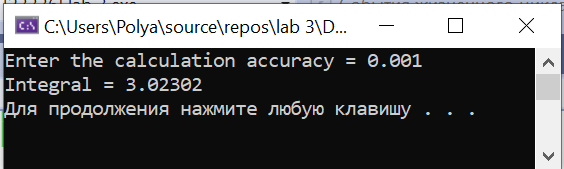
} while (fabs(s - s0) >= e);

Res = s;

cout << "Integral = " << Res << endl;

system("pause");

}



**Висновок:** в процесі виконання роботи я закріпила свої знання про особливості організації ітераційних циклів. Дані цикли використовуються, коли кількість повторень невідома. Існує 2 різновиди ітераційних циклів: цикли з передумовою (перевірка умови повторення циклу перед кожною його ітерацією; цикл може не виконуватись жодного разу) і цикли з постумовою (перевірка умови повторення циклу після виконання тіла циклу; тіло циклу виконується як мінімум один раз). Цикли з передумовою реалізуються за допомогою while, з постумовою – do-while (у Python подібний оператор відсутній, тому цикли з постумовою реалізуються за допомогою if). В нашій задачі ми використали цикл з постумовою.