# Міністерство освіти і науки України

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

Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 6 з дисципліни

«Основи програмування-1.

Базові конструкції»

«Організація підпрограм»

Варіант 20

Виконав студент Лошак Віктор Іванович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота 6**

**Мета:**

Набути навичок складання і використання підпрограм користувача

**Тема:**

Задане дійсне число у, отримати:

Де g(x)=

**Математична модель:**

Для обчислення виразу при заданому y використаємо підпрограму. Значенням у є число що вводиться користувачем за запитом на початку програми. Для спрощення обрахунків використаємо дві підпрограми. Перша підпрограма буде розраховувати значення факторіалу переданого їй параметру. Друга підпрограма буде розраховувати значення функції g(x) де х- переданий їй аргумент. Аргументи передаватимемо по значенню, оскільки вони мають цілий тип даних. Для обчислення значень змінних всередині функцій скористаємося фунціоналом сторонніх бібліотек. Для використання математичних функцій степеня використаємо бібліотеки <cmath> для С++ та імпортуємо math для пайтон. Для перевірки користувацького вводу використаємо логічні оператори. Для обрахування суми в функціях скористаємося арифметичними циклами.

**1)**Приклад коду на С++:

#include <iostream>

# include <cmath>

using namespace std;

long long int factorial(long int value)

{

long long fact = 1;

if (value == 0)

return 1;

else if (value > 0)

{

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

{

fact \*= i;

}

return fact;

}

else if (value < 0)

{

cout << " Can`t get factorial of this number.\n";

exit(-1);

}

}

double function(double x)

{

long double numerator=0, denominator=0;

for (int k = 0; k < 5; ++k)

{

numerator += pow(x, (2 \* k) + 1)/factorial(2\*k+1);

denominator += pow(x, 3 \* k) / factorial(3 \* k);

}

return (numerator / denominator);

}

void main( )

{

cout << "Enter the y value please: ";

double y;

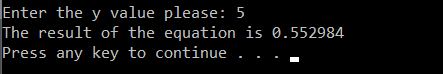
cin >> y;

double result = (1.7 \* function(0.25) + 2 \* function(1 + y)) / (6 - function(pow(y, 2) - 1));

cout << "The result of the equation is " << result << endl;

}

Виконання коду на С++:



**2)**Приклад коду на Python:

import os

import math

def factorial(value: int):

    fact = 1

    if value == 0:

        return 1

    elif value > 0:

        for i in range(1, value+1):

            fact \*= i

        return fact

    elif value < 0:

        print(" Can`t get factorial of this number.\n")

        exit(-1)

def function(x: float):

    numerator=float(0)

    denominator=float(0)

    for k in range(0,5):

        numerator += pow(x, (2 \* k) + 1)/factorial(2\*k+1)

        denominator += pow(x, 3 \* k) / factorial(3 \* k)

    return (numerator / denominator)

def main():

    y=float(input("Enter the y value please: "))

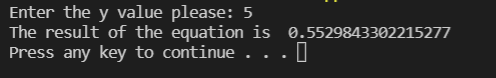
    result = float((1.7 \* function(0.25) + 2 \* function(1 + y)) / (6 - function(pow(y, 2) - 1)))

    print("The result of the equation is ",result)

main()

os.system("pause")

Виконання коду на Python:



**Висновок:**

Отже за допомогою даного алгоритму ми успішно обрахували і відобразили на екрані значення заданого виразу при довільних ввідних даних від користувача, за допомогою використання підпрограм на компільованій мові С++ та інтерпретованій Python. Ми також створили механізми валідації користувацького вводу та аналізували задане рівняння для знаходження оптимальних шляхів виконання завдання.