

Додаток 1

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 № 3 з дисципліни
«Основи програмування 1. Базові конструкції»

«Організація розгалужених процесів»

Варіант №34

Виконав студент Щербацький Антон ІП-14
(шифр, прізвище, ім'я, по батькові)

Перевірив Камінська Поліна Анатоліївна
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №3

Варіант 34

Мета: вивчити особливості організації ітераційних циклів.

Хід роботи

Постановка задачі

34. З точністю $\epsilon = 10^{-8}$ обчислити значення функції $\frac{e^x - e^{-x}}{2}$ за формулою:

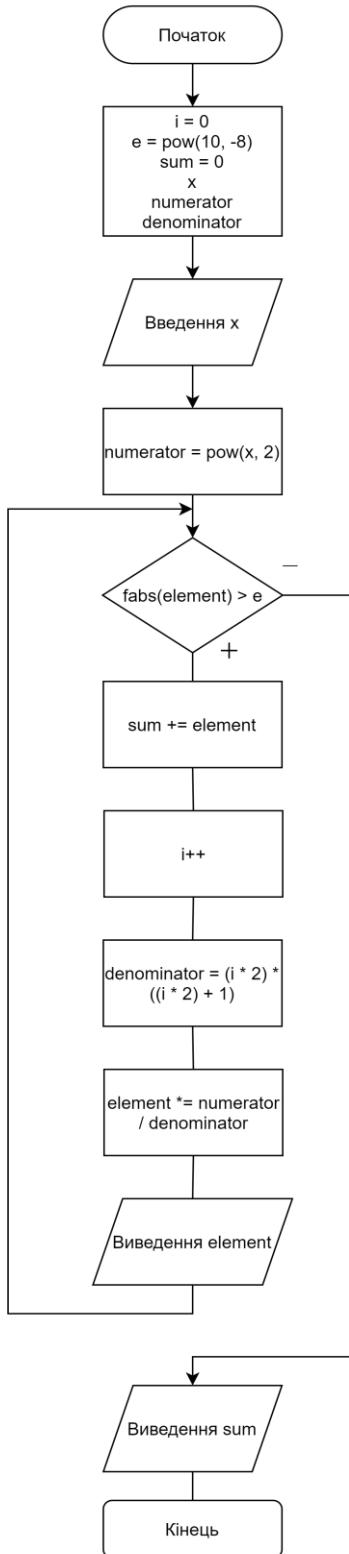
$$S = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \dots,$$

використавши рекурентну формулу для обчислення члена ряду.

Математична модель

Змінна	Тип	Ім'я	Призначення
Аргумент	Дійсний	x	Вхідне дане
Точність	Дійсний	e	Проміжні дані
Елемент рекурентної формули	Дійсний	Element	Проміжні дані
Номер ітерації	Натуральний	i	Проміжні дані
Чисельник дробу	Дійсний	numerator	Проміжні дані
Знаменник дробу	Дійсний	denominator	Проміжні дані
Значення функції	дійсний	sum	Результат

Блок схема



Випробування коду на C++:

Код:

```
1 #include <iostream>
2 #include <cmath>
3 #include <iomanip>
4 using namespace std;
5
6 ► int main()
7 {
8     int i = 0;                                //змінна для
9     double e = pow( _Left: 10, _Right: -8), sum = 0, x, numerator, denominator; //x - вхідне
10
11    cout << "Enter x: " << fixed << setprecision(10) << endl; cin >> x;      //вводимо з
12
13    double element = x;                      //ініціалізація
14    numerator = pow(x, _Right: 2);
15
16    while (fabs(element) > e)
17    {
18        sum += element;                      //знаходження суми
19        i++;
20        denominator = (i * 2) * ((i * 2) + 1);
21        element *= numerator / denominator;
22        cout << i << " Element = " << element << endl;
23    }
24    cout << "Sum = " << sum;
25 }
```

Результат:

При $x = 10$

```
Enter x:  
10  
1 Element = 166.6666666667  
2 Element = 833.333333333  
3 Element = 1984.1269841270  
4 Element = 2755.7319223986  
5 Element = 2505.2108385442  
6 Element = 1605.9043836822  
7 Element = 764.7163731820  
8 Element = 281.1457254346  
9 Element = 82.2063524662  
10 Element = 19.5729410634  
11 Element = 3.8681701706  
12 Element = 0.6446950284  
13 Element = 0.0918368986  
14 Element = 0.0113099629  
15 Element = 0.0012161250  
16 Element = 0.0001151634  
17 Element = 0.0000096776  
18 Element = 0.0000007265  
19 Element = 0.0000000490  
20 Element = 0.0000000030  
Sum = 11013.2328747002  
Process finished with exit code 0
```

При x = 0

```
Enter x:  
0  
Sum = 0.0000000000  
Process finished with exit code 0
```

При x = -15

```
1 Element = -562.5000000000
2 Element = -6328.1250000000
3 Element = -33900.6696428571
4 Element = -105939.5926339286
5 Element = -216694.6212966720
6 Element = -312540.3191778924
7 Element = -334864.6276905990
8 Element = -277001.9898175910
9 Element = -182238.1511957836
10 Element = -97627.5809977412
11 Element = -43411.4737638177
12 Element = -16279.3026614316
13 Element = -5217.7252119973
14 Element = -1445.7982422406
15 Element = -349.7898973163
16 Element = -74.5290974395
17 Element = -14.0916360705
18 Element = -2.3803439308
19 Element = -0.3613882486
20 Element = -0.0495807048
21 Element = -0.0061769981
22 Element = -0.0007019316
23 Element = -0.0000730502
24 Element = -0.0000069882
25 Element = -0.0000006166
26 Element = -0.0000000503
27 Element = -0.0000000038
Sum = -1634508.6862358977
```

Висновок

На цій лабораторній роботі було досліджено та набуто навичок подання операторів повторення, також побудовано математичну модель, блок-схему. В результаті виконання цієї роботи було досліджено та отримано відповідне значення суми за допомогою рекурентної формули для обчислення n-ого члена числового ряду із заданою точністю.