**Звіт виконання лабораторної роботи з теми Циклічні обчислювальні процеси**

*Виконав студент групи ІС-21, ФІОТ*

*Коваленко Дмитро Андрійович*

*Варіант № 10*

**Мета**

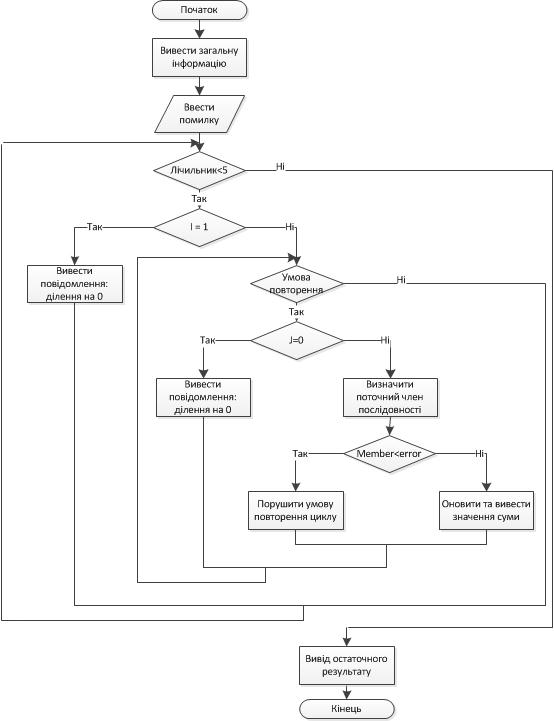
1. Вивчити особливості циклічних обчислювальних процесів
2. Опанувати технологію використання операторів циклів
3. Навчитися розробляти алгоритми та програми циклічних процесів

**Завдання**

Обчислити вираз за приведеною формулою.

**C:\Users\Dima\Desktop\var10.gif**

**Блок-схема алгоритму**

****

**Код програми**

#include <iostream>

#include <iomanip>

#include <math.h>

using namespace std;

double memberVal(double x, double k); // Calculating expression

void sum(double error); // Summing elements

void main()

{

double error; // Error of the calculations ( the programm will end calculations if each member is less than error)

cout<<"Dima Kovalenko IS-21\n";

cout<<"Calculating expression: \n";

cout<<"Sum(x from 1 to 5)(Sum (k from 0 to infinity) ( (-1)^k\*sqrt(x+1)/(1-x^k)^2))\n\n";

cout<<"Enter error: ";

cin>>error;

sum(error); //executing the sum function

system("pause");

}

double memberVal(double x, double k) // Calculating expression : pow(-1,k)\*sqrt(x+1)/((1-pow(x,k))\*(1-pow(x,k)))

{

return pow(-1,k)\*sqrt(x+1)/((1-pow(x,k))\*(1-pow(x,k)));

}

void sum(double error) //Summing elements

{

cout<<"=================================================="<<endl; // displaying header of the table

cout<< setw(5)<< "x" <<setw (10)<< "k" << setw (20) << "Member" << setw (15) << "Result" << endl;

cout<<"=================================================="<<endl;

double result = 0; // Result of summing

for (int i = 1; i <= 5; i++) // Cycle for calculating outer summ

{

if ( i == 1) // Avoiding problem: dividing by zero

{

cout<<"Error:"<<setw(44)<< " Dividing by zero, skipping element;\n";

continue;

} else

{ bool check=0;

for (int j = 0;check==0 ; j++) //Cycle for calculating inner summ

{ check=0;

if ( j == 0) // Avoiding problem: dividing by zero

{

cout<<"Error:"<<setw(44)<< " Dividing by zero, skipping element;\n";

continue;

} else

{

double member; // Each member of the sequence

member=memberVal(i,j);

if (fabs(member) < fabs(error)) check=1; // If member is less then error , programm breaks the cycle

result+= member;

cout<< setw(5)<< i <<setw (10)<< j << setw (20) << member << setw (15) << result << endl;

}

}

}

}

cout<<"=================================================="<<endl;

cout.precision(9);

cout<<"Summ = "<<result<<endl;// displaying result

}

**Screen Shot результатів**

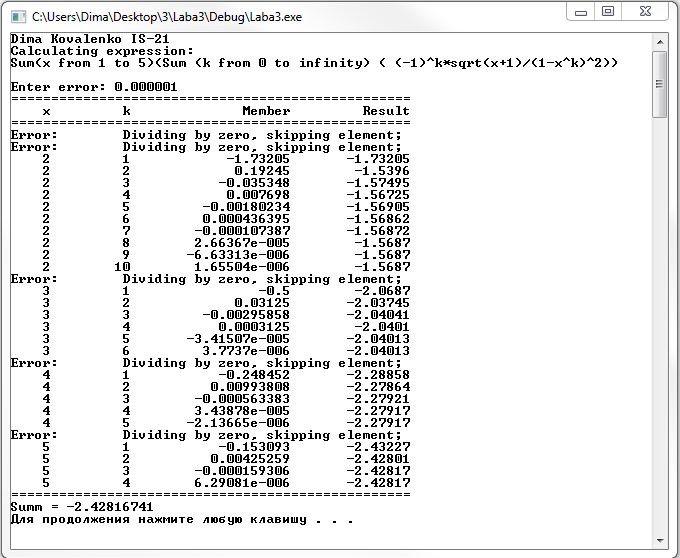
****

Рис.1 Таблиця, отримана при введенні значення error = 0.000001

**Аналіз роботи**

Дана таблиця Excel показує поточні значення виразу та рахує їх суму.

Значення суми в програмі та в таблиці збігаються, отже програма працює правильно.

Дана таблиця також показує місця переходу до наступної ітерації зовнішнього циклу.

|  |  |  |  |
| --- | --- | --- | --- |
| member=(-1)^k\*sqrt(x+1)/(1-x^k)^2 | | | |
|
| Error = 1E-6  (елементи послідовності які менші за данне число враховуватися не будуть) | | | |
|
|
| **x** | **k** | **member** |  |
| **2** | **1** | **-1,732050808** |  |
| **2** | **2** | **0,19245009** |  |
| **2** | **3** | **-0,035347976** |  |
| **2** | **4** | **0,007698004** |  |
| **2** | **5** | **-0,001802342** |  |
| **2** | **6** | **0,000436395** |  |
| **2** | **7** | **-0,000107387** |  |
| **2** | **8** | **2,66367E-05** |  |
| **2** | **9** | **-6,63313E-06** |  |
| **2** | **10** | **1,65504E-06** |  |
| **2** | **11** | **-4,13357E-07** | **Число не враховується** |
| **Перехід до наступної ітерації зовнішнього циклу** | | |  |
| **3** | **1** | **-0,5** |  |
| **3** | **2** | **0,03125** |  |
| **3** | **3** | **-0,00295858** |  |
| **3** | **4** | **0,0003125** |  |
| **3** | **5** | **-3,41507E-05** |  |
| **3** | **6** | **3,7737E-06** |  |
| **3** | **7** | **-4,18533E-07** | **Число не враховується** |
| **Перехід до наступної ітерації зовнішнього циклу** | | |  |
| **4** | **1** | **-0,248451997** |  |
| **4** | **2** | **0,00993808** |  |
| **4** | **3** | **-0,000563383** |  |
| **4** | **4** | **3,43878E-05** |  |
| **4** | **5** | **-2,13665E-06** |  |
| **4** | **6** | **1,33345E-07** | **Число не враховується** |
| **Перехід до наступної ітерації зовнішнього циклу** | | |  |
| **5** | **1** | **-0,153093109** |  |
| **5** | **2** | **0,004252586** |  |
| **5** | **3** | **-0,000159306** |  |
| **5** | **4** | **6,29081E-06** |  |
| **5** | **5** | **-2,50988E-07** | **Число не враховується** |
| **СУММА** | | **-2,42816741** |  |

**Висновок**

Програма працює правильно, про що свідчить аналіз результатів та ScreenShot’и. Використано цикли з постумовою, цикли for та розгалуження.