Національний технічний університет України

«Київський політехнічний інститут»

Кафедра автоматизованих систем обробки інформації та управління

Звіт про виконання лабораторної роботи №4 з дисципліни

«Алгоритмизація та програмування»

Перевірено: Ковалюк Т.В. Виконав ст. 1 курсу ФОІТ

Проскура С.Л. гр. ІС-52

Дорошенко А.В.

Київ 2015

## Лабораторна робота 4

# **Цикли з розгалудженням**

Варіант № 9

*Мета:*

1. Вивчити особливості циклічних обчислювальних процесів з розгалуженнями
2. Опанувати технологію рекурентних обчислень
3. Навчитися розробляти алгоритми та програми розвинення функцій у ряди

### Завдання

Обчислити значення функції у, розвинувши функцію ln(x) у ряд Тейлора. Аргумент х змінюється від -2 до 2 з кроком 0.5. Визначити похибку.

|  |  |
| --- | --- |
|  |  |

C:\Users\Anton\AppData\Local\Temp\Rar$EXa0.257\MethodLabWorkC++Semestr1Html\labs\lab_04\pictures\var9.gif

Значення похибки задати самостійно.

### Блок-схема алгоритму



Рис.1 Алгоритм обчислення натурального логарифму за формулою Тейлора

Рис.2 Головний алгоритм

### Код програми

//Laboratory work

//written by student of the first curse

//of the group IC-52

//Anton Doroshenko

//2015.11.01

//==========================================================================

#include <iostream>

#include <math.h>

#include <iomanip>

using namespace std;

double t = 0; //точність розрахунків

double ln(double a)

{

int i = 2; //лічильник членів ряду

double member = a - 1; //перший член ряду

double sum = member; //сума ряду

while (fabs(member) > t) //доки поточний член не задовольняє точності

{

member \*= -(a - 1); //обчислювати поточний член

sum += member / i; //і суму ряду

i++;

if(i == 1000)

break;

}

return sum; //повернення значення натурального логарифму

}

int main()

{

cout << "laboratory work number 4 made by Anton Doroshenko, IS-52" << endl;

bool flag = true;

double x,teylorsum,standart; //аргумент, вираз за формулою Тейлора,значення стандартної функції log()

double xstart, step, xend; //початкове значення аргументу,крок зміни, кінцеве значення аргументу

double error; //похибка

int cont;

do

{

cout << "Enter xstart, xend and step (-2 2 0.5)" << endl;

cin >> xstart >> xend >> step;

cout << "Input t" << endl;

cin >> t;

cout<<setw(5)<<"x"<<setw(20)<<"teylorsum"<<setw(25)<<"standart"<<setw(20)<<"error"<<endl;

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

for (x = xstart; x <= xend; x += step) //цикл перебору значень аргументів функції

{

if (x > 0 && x <= 1)

{

teylorsum = ln(x / 2) - 1; //визначення функції за формулою Тейлора

standart = log(x / 2) - 1; //визначення функції за стандартною формулою

error=fabs(standart-teylorsum);//розбіжності між точним і наближени значенням

flag = true; //функція визначена

}

else

flag = false; //функція не визначена

if (flag == true)

{

cout<<setw(5)<<x<<setw(20)<<teylorsum<<setw(25)<<standart<<setw(20)<<error <<endl;

cout << " " << endl;

}

else

{

cout << setw(5) << x << setw(35) << "not defined" << endl;

cout << " " << endl;

}

}

cout << "Enter 1 if you want to start again" << endl;

cin >> cont;

} while (cont == 1);

}

1. **Screen Shot результатів**

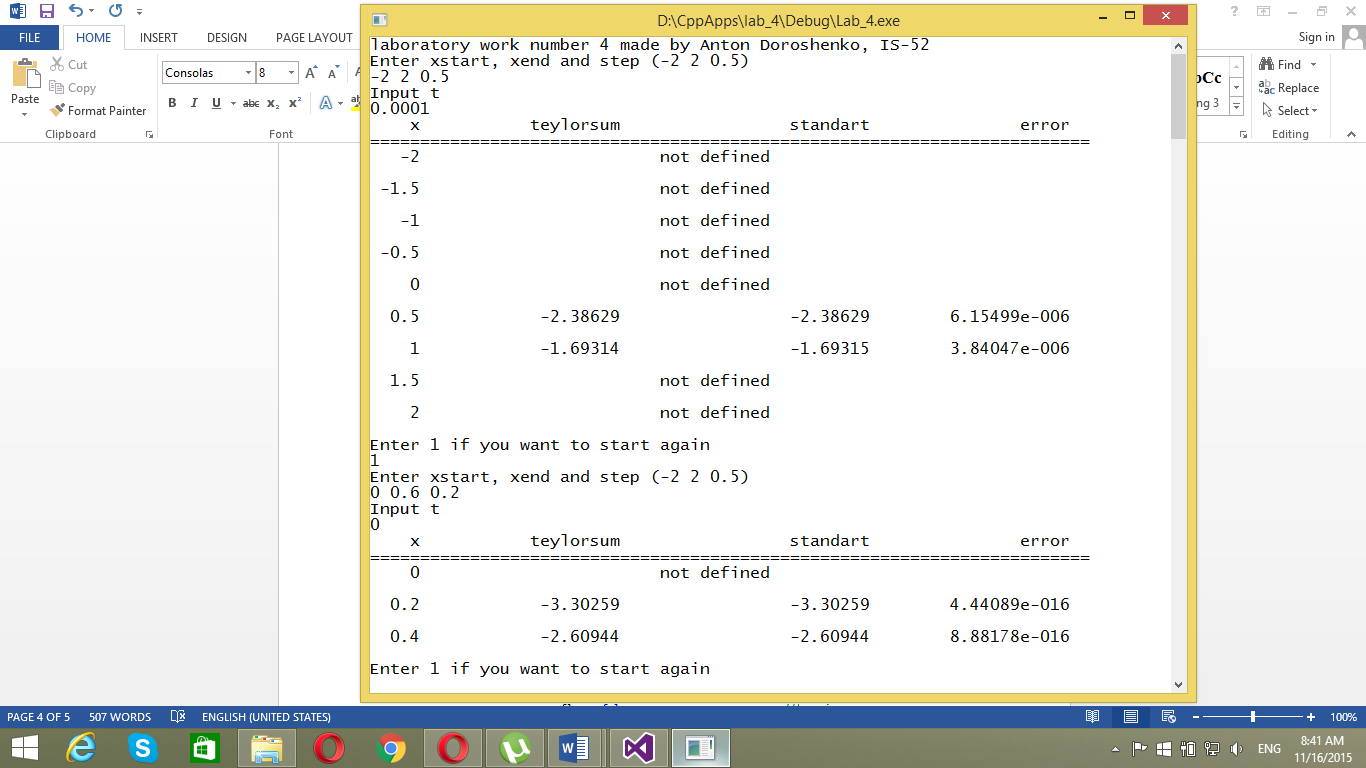


Рис.3 Screen Shot результатів

1. **Аналіз результатів**

Як ми бачимо зі ScreenShot’ів, ми ввели 2 варіанти даних.

1. При аргументі меншому або рівного 0 і більшого 1 функція невизначена.
2. При інших значеннях аргумента обчислюється вираз за допомогою функції Тейлора, вираз за допомогою стандартної функції та похибка.
3. **Висновок**

Програма працює правильно, про що свідчить аналіз результатів та ScreenShot’и. Програма захищена від взяття логарифма не додатнього числа. Програма коректно виводить проміжні результати. Використано алгоритм розвинення функції у ряд Тейлора.