МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКО”

Факультет прикладної математики

Кафедра системного програмування і спеціальних комп’ютерних систем

Лабораторна робота №1

З дисципліни «Алгоритми та методи обчислень»

«Обчислення значень функції»

Виконав:

студент III-го курсу

групи КВ-41

Горпинич-Радуженко Іван

Київ 2016

Варіант 5:
Функція: *shx*; Інтервал [−9.8; 13.9];

Завдання для лабораторної роботи:

1. Побудувати таблицю залежності довжини ряду *n*, що забезпечує точність функції не меншу за задане значення *eps* у точці $x = (b + a)/2$, від *eps*:

eps	*n*	Абсолютна похибка	Залишковий член
10^{-2}	4	0.005	0.001
...	...	...	...

Значення *eps* змінюється від 10^{-2} до 10^{-14} з кроком 10^{-3}.

2. Для *n* (довжина ряду фіксована й дорівнює *n*), отриманого в п.1 при $eps = 10^{-8}$, у точках $xi = a + h*i$, $h = (b – a)/10$, $i = 0, ..., 10$ обчислити абсолютну похибку та залишковий член ряду. Результати подати у вигляді таблиці:

x_i	Абсолютна похибка	Залишковий член
0	0.005	0.001
...	...	...

3. За допомогою AdvancedGrapher побудувати графік залежності абсолютної похибки від *x* (у логарифмічному масштабі).

Текст програми:

Main.cpp:

```
#include <iostream>
#include <fstream>
#include "sinhx.h"

using namespace std;

int main() {
	ofstream tbl("table.csv");
	const double a = -9.8;
	const double b = 13.9;
	const double h = (b-a)/10;
	double eps;
	double x = (a+b)/2;
	Sinh *sinhx = new Sinh();
	Result result;

	cout << "\t\t TABLE 1\t" << endl;
```

```
	cout<<'|'<<" Eps\t"<< '|' <<"  n\t"<< '|' <<" Absolute Error "<< '|' <<" Remainder term"<<endl;
	cout << "--------------------------------------------------"<<endl;

	for(eps = 1e-2; eps >= 1e-14; eps *= 1e-3)
	{
		result = sinhx->AccuracyValue(x, eps);
		cout<< '|' <<eps<<"\t"<< '|' <<"  "<<result.n<<"\t"<< '|' <<" "<<result.absEr<<"\t "<< '|' <<result.remT<<endl;
	}
	cout<<"\n"<<endl;

	int n = sinhx->AccuracyValue(x, 1e-8).n;
	cout << "\t\t TABLE 2\t" << endl;
	cout<< '|' <<"  Xi\t"<< '|' <<" Absolute Error"<< '|' <<" Remainder term"<<endl;
	cout << "-----------------------------------------" << endl;
	for (int i = 0; i <= 10; ++i)
	{
		x = a + h*i;
		result = sinhx->AbsoluteError(x, n);
		tbl << x << ';' << result.absEr << ';' << endl;
		cout<< '|' <<x<<"\t"<< '|' <<" "<<result.absEr<<"\t"<< '|' <<result.remT<<endl;
	}
	system("PAUSE");
	return 0;
}
```

Result.h:

```
struct Result
{
	double f_x;
	int n;
	double absEr;
	double remT;

	Result();
	~Result();
	Result& operator=(Result& src);
```

Result.cpp:

```
#include "Result.h"

Result& Result::operator=(Result& src)
{
	if (this == &src) return src;
	f_x = src.f_x;
	n = src.n;
	absEr = src.absEr;
	remT = src.remT;

	return src;
}

Result::~Result() {};
Result::Result() {};
```

Sinhx.h:

```
#pragma once
#define _USE_MATH_DEFINES

#include <cmath>
#include "Result.h"
```

```
class Sinh
{
private:
	int sign;

public:
	Sinh();
	~Sinh();
	Result AccuracyValue(double x, double eps);
	Result AbsoluteError(double x, int n);
};
```

Sinhx.cpp:

```
#include "sinhx.h"
#include <iostream>

Sinh::Sinh() {}
Sinh::~Sinh() {}

Result Sinh::AccuracyValue(double x, double eps) {
	double U, result = 0;
	int k;
	double lib_sin = sinh(x);
	Result Res;

	U = x;
	for(k = 1; abs(U) >= eps; ++k)
		{
			result += U;
			U *= x*x/(2*k * (2*k + 1));
		}


	Res.absEr = abs(result - lib_sin);
	Res.n = k;
	Res.remT = U;
	Res.f_x = result;
	return Res;
}

Result Sinh::AbsoluteError(double x, int n) {
	double U, result = 0;
	double lib_sin = sinh(x);
	int k = 1;
	Result Res;

	Res.remT = 0;

	if(n > 0)
	{
		U = x;
		x *= x;

			while(--n)
			{
				result += U;
				U *= x/(2*k * (2*k + 1));
				k++;
			}


		Res.remT = U;
```

```
        }
        Res.absEr = abs(result - lib_sin);
        Res.n = k;
        Res.f_x = result;

        return Res;
}
```

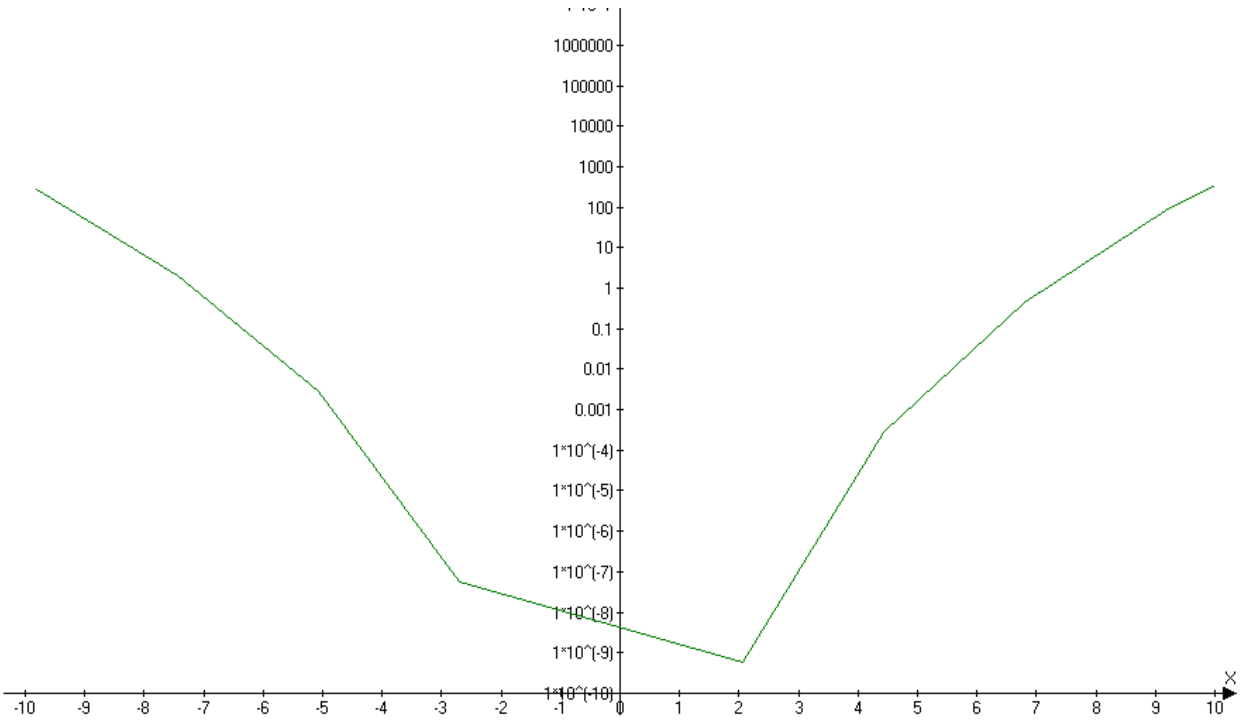
Вихідні данні:

TABLE 1

Eps	n	Absolute Error	Remainder term
0.01	5	0.00183123	0.00176206
1e-05	7	1.85037e-06	1.81351e-06
1e-08	9	5.67681e-10	5.60721e-10
1e-11	10	6.95932e-12	6.89015e-12
1e-14	12	4.44089e-16	5.72591e-16

TABLE 2

Xi	Absolute Error	Remainder term
-9.8	271.11	-199.423
-7.43	2.13563	-1.80194
-5.06	0.00283649	-0.00262718
-2.69	5.80725e-08	-5.68487e-08
-0.32	0	-1.08763e-23
2.05	5.67681e-10	5.60721e-10
4.42	0.000279492	0.0002637
6.79	0.448557	0.389689
9.16	82.481	63.2637
11.53	4910.9	3162.49
13.9	149308	75889.7



Висновки:

В ході виконання лабораторної роботи ми обчисляли наближене значення функції sh(x) використовуючи розкладання в ряд Маклорена.

Перше завдання стосувалося побудови таблиці залежності довжини ряду, що забезпечує точність функції не меншу за задане значення eps.

Отриманні результати свідчать про те, що зі збільшенням точності(зменшенням значення допустимої похибки) збільшується кількість членів ряду Маклорена, необхідна для отримання результату з заданою точністю.

Друге завдання стосувалося обрахунку залежності абсолютної похибки від значення аргументу при заданій довжині ряду. Характер отриманої залежності є таким, що при збільшенні значення функції збільшується абсолютна похибка.

Це зумовлено тим, що заданої точності недостатньо (тому що при збільшенні значення функції гіперболічного синусу, значення аргументу зростає у квадратичній прогресії).