Лабораторна робота 7

Завдання до лабораторних робіт з курсу «Програмування» (1 курс)

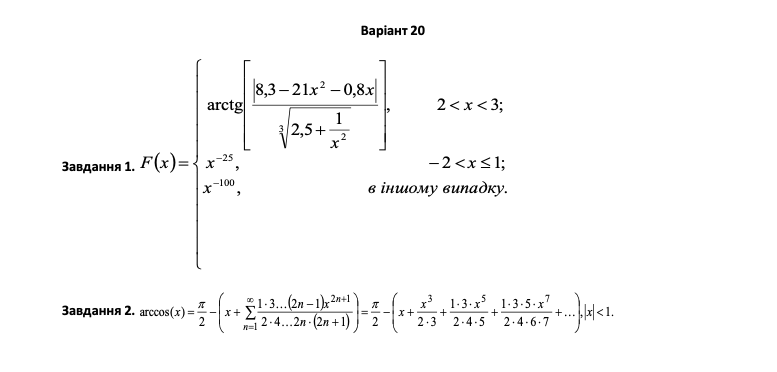
Завдання:

Тема. : "Програмування ітераційних процесів та створення функцій".

1. Складіть програму, яка виводить на екран значення функції F(x) у заданому діапазоні. Області допустимих значень параметрів формул визначте самостійно. Діапазон і крок зміни аргументу користувач задає у командному рядку; в тому разі, якщо в командному рядку параметри не задано, або задано їх у недостатній кількості, програма пропонує здійснити ввід з клавіатури під час виконання програми.

2. Складіть програму обчислення функції, заданою за допомогою ряду. Необхідні дані введіть з клавіатури під час виконання програми. Виведіть на екран:

* −  суму перших 5 членів ряду;
* −  суму ряду із введеною точністю ε та кількість виконаних ітерацій;
* −  значення функції, обчислене за допомогою стандартних математичних функцій.

3. За результатами виконання лабораторної роботи створіть звіт в електронному та друкованому вигляді (вимоги до звіту додаються). 

**Завдання №1**Щоб отримати значення a,b,h в числовому типі я використав конструкцію з stringstream

Щоб конвертувати значення в масиві типу char в тип double. Далі йдуть перевірки на кількість введених аргументів і виклик функції prog1

Функція Prog1:

Ця функція використовується для перевірки на невід’ємність значень, виклику функції calculate та виводу результату на екран. Примітка\* в циклі зі значеннями double не вдається зрівняти значення з 0, тому я вирішив задати умову, що якщо число входить до області -0.0000001<i<0.000000001 то воно є 0 (це потрібно тому що треба виключити 0 з ОДЗ)

Функція calculate:

Використовується для обчислення заданої функції враховуючи ОДЗ для кожного рівняння

**Код програми**

#include <iostream>

#include <cmath>

#include <iomanip>

#include <sstream>

**using** **namespace** std;

**double** calculate(**double** x){

**double** y;

**if**(x>2 && x<3){

**double** denominator,numerator;

denominator=8.3-21\*x\*x-0.8\*x;

denominator=(denominator < 0) ? -denominator:denominator;

numerator=2.5+1/(x\*x);

numerator=pow(numerator, 1.0 / 3);

y=**int**(denominator/numerator);

y=atan(y);

}**else** **if**(x>-2 && x<=1){

y=pow(x, -25);

}**else**{

y=pow( **double**(x), -100);

}

**return** y;

}

**void** prog1(**double** a, **double** b, **double** h){

**if**(a>b){

**double** t=a;

a=b;

b=t;

}

**if** (h<0)

h=-h;

**double** i=a;

cout.precision(5);

**while** (i<=b) {

**if**(!(i>-0.0001 && i<0.0001)){

cout<<scientific<<"x="<<i<<"\t\t y=";

cout<<calculate(i)<<endl;

}**else**{

cout<<"x="<<0<<"\t\t\t\t\t y="<<"Не Існує"<<endl;

}

i=i+h;

}

}

**int** main(**int** argc, **char** \* argv[]) {

**double** a{},b{},h{};

**if**(argc==4){

stringstream converta{ argv[1] };

stringstream convertb{ argv[2] };

stringstream converth{ argv[3] };

**if** (!(converta >> a)) // do the conversion

a = 0;

**if** (!(convertb >> b)) // do the conversion

b = 0;

**if** (!(converth >> h)) // do the conversion

h = 0;

}**else**{

cout<<"Введіть інтервал [a,b] і крок h змінювання x"<<endl;

**if**(argc==3){

stringstream converta{ argv[1] };

stringstream convertb{ argv[2] };

**if** (!(converta >> a)) // do the conversion

a = 0;

**if** (!(convertb >> b)) // do the conversion

b = 0;

cout<<"h=";cin>>h;

}**else** **if**(argc==2){

stringstream converta{ argv[1] };

**if** (!(converta >> a)) // do the conversion

a = 0;

cout<<"b=";cin>>b;

cout<<"h=";cin>>h;

}**else** **if**(argc==1){

cout<<"a=";cin>>a;

cout<<"b=";cin>>b;

cout<<"h=";cin>>h;

}

}

prog1(a,b,h);

}

**Керівництво користувача**

Програму потрібно запускати командою ./” File Name ” [arg1 arg2 arg3]

Все що в [ ] – необов’язково

Якщо користувач не введе 3 аргументи то йому запропонують ввести їх потім  
Якщо ж він порушить правила ОДЗ то програма повідомить йому про це .

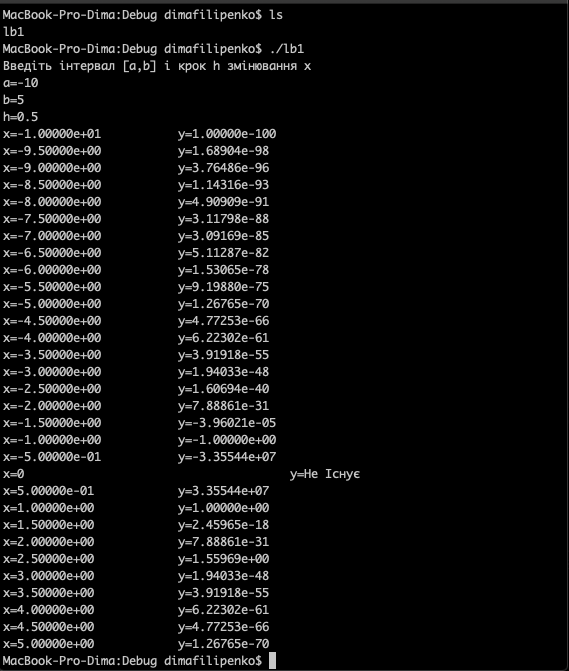
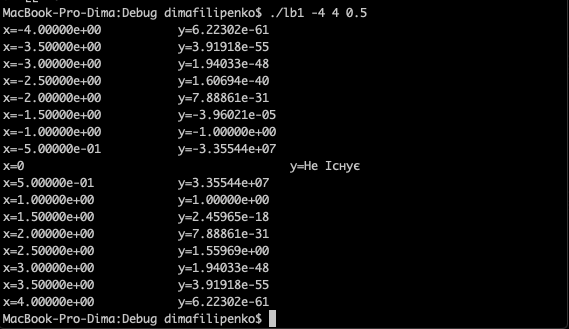
Ви вводите 3 значення [a,b] та h, де [a,b] – проміжок функції на осі x h- крок змінювання x

A,b можна вводити в будь-якому порядку, вони будуть розташовані в порядку зростання

h може бути дійсним числом   
**Тести**

****

****

****

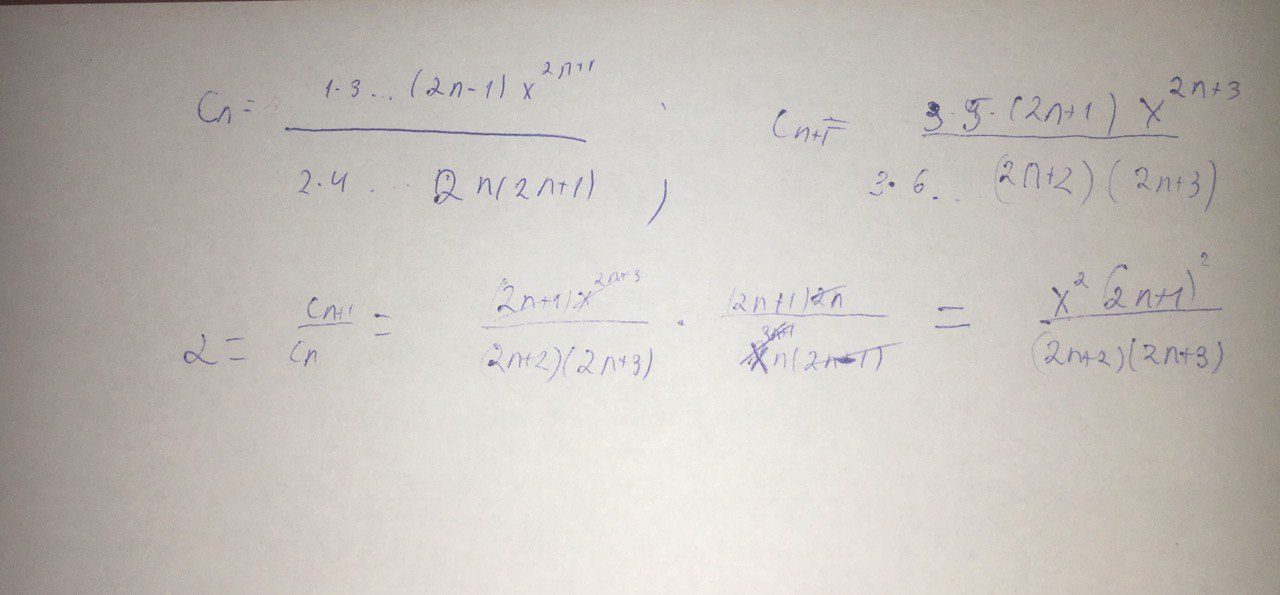
**Завдання №2**Щоб отримати значення x,E в числовому типі я використав конструкцію з stringstream

Щоб конвертувати значення в масиві типу char в тип double. Далі йдуть перевірки на кількість введених аргументів і виклик функції prog2

Якщо ми маємо 2 перших відомих члена ряду, то ми можемо знайти їх частку .  
І в подальшому використовувати цю частку. В циклі можна множити її на член ряду і отримувати наступний при цьому складуючи всі члени ряду в суму .  
Чим більше елементів в цьому ряді тим краща буде точність, Тому в циклі з умовою поки член ряду>E ми постійно отримуємо наступний член та складуємо їх

Для перевірки правильності

Частка Cn та Cn+1



**Код програми**

#include <iostream>

#include <cmath>

#include <iomanip>

#include <sstream>

**using** **namespace** std;

**void** wait\_key(){

**do**

{

cout << '\n' << "Press a key to continue...";

} **while** (cin.get() != '\n');

getchar();

}

**void** prog2(**double** x,**double** E){

**int** n;

**double** S=0,result,alf;

**int** count=0;

cout.precision(15);

**double** part=pow(x,3)/6;

S+=part;

**for**(n=1;n<=5;n++){

alf=x\*x\*(2\*n+1)\*(2\*n+1)/( (2\*n+2)\*(2\*n+3) );

part=part\*alf;

S+=part;

}

result=M\_PI\_2-(x+S);

cout<<"Сума перших 5 членів ряду становить"<<endl<<"S="<<result<<endl;

S=0;

n=1;

part=pow(x,3)/6;

S+=part;

**while** (abs(part)>abs(E)) {

alf=x\*x\*(2\*n+1)\*(2\*n+1)/( (2\*n+2)\*(2\*n+3) );

part=part\*alf;

S+=part;

n++;

count++;

}

result=M\_PI\_2-(x+S);

cout<<"Сума ряду з точністью "<<E<<" становить"<<endl<<"S="<<result<<endl;

cout<<"Кількість виконаних ітерацій: "<<count<<endl;

cout<<"Резульатат обчисленний за допомогою стандартних математичних функцій "<<acos(x)<<endl;

wait\_key();

}

**int** main(**int** argc, **const** **char** \* argv[]) {

**double** x{},E{};

**if**(argc==3){

stringstream convertx{ argv[1] };

stringstream convertE{ argv[2] };

**if** (!(convertx >> x)) // do the conversion

x = 0;

**if** (!(convertE >> E)) // do the conversion

E = 0;

}**else** **if**(argc==2){

stringstream convertx{ argv[1] };

**if** (!(convertx >> x)) // do the conversion

x = 0;

cout<<"E=";cin>>E;

}**else** **if**(argc==1){

cout<<"\_\_\_\_\_\_\_\_\_\_\_Лабораторна №7 завадння 2\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl<<endl;

cout<<"введіть число |x|<1 та точність E"<<endl;

cout<<"x=";cin>>x;

cout<<"E=";cin>>E;

}

**while**(abs(x)>1 || x==1){

cout<<"Помилка. Потрібно щоб виконувалася нерівність |x|<1 "<<endl;

cout<<"x=";cin>>x;

}

cout<<"Результат обчислень:"<<endl<<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"<<endl;

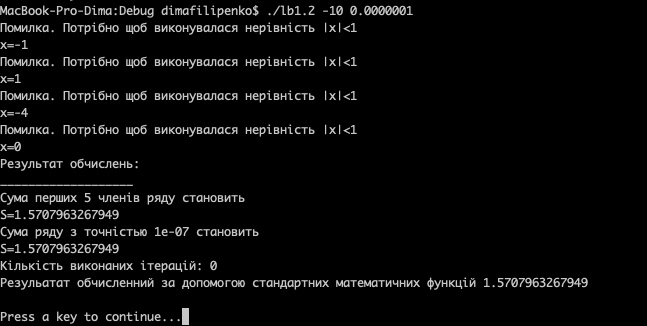
prog2(x,E);

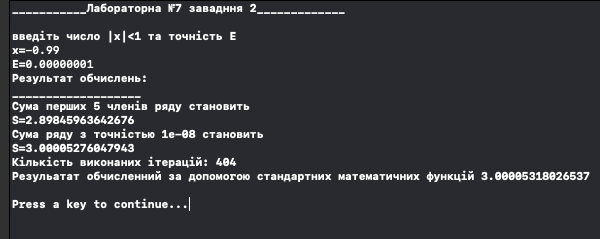
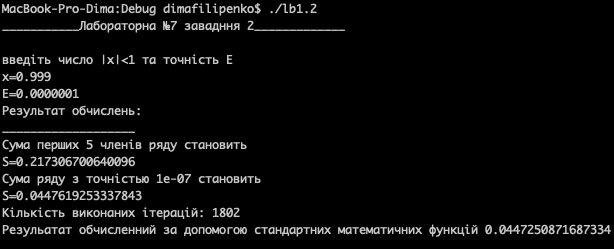
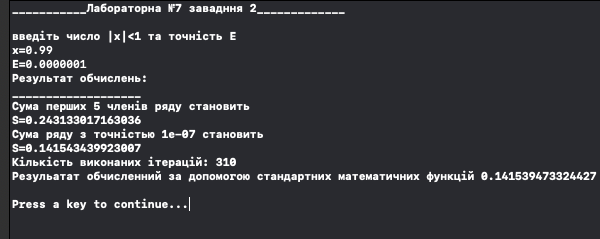
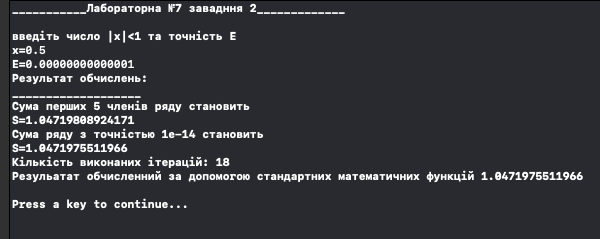
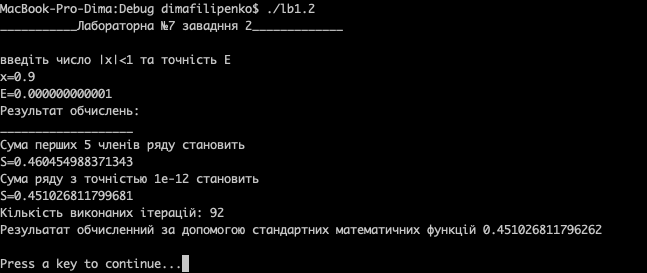
}

**Керівництво користувача**При запускі програми потрібно ввести число x для функції та E для більшої точності при розрахунках. В результаті потрібно отримати близьке значення до стандартної математичної функції.

Якщо |x|<1 виведеться відповідне повідомлення

**Тести**

****

****

**Висновки**

*Після цієї лабораторної роботи я дізнався про реалізацію передавання аргументів через командний рядок, а також про те як можна розкласти математичну функцію на ряд Тейлора, і знайти її значення Оптимізованим шляхом і з високою точністю*