

ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №6

за курсом “Програмування”

студентки групи ПА-23-1

Мазур Віолети

кафедра комп'ютерних технологій, ДНУ

2023/2024

1. Постановка задачі

1. Складіть програму, яка виводить на екран таблицю значень функції $F(x)$ у заданому діапазоні. Значення функції $F(x)$ обчисліть за формулою, наведеною у вашому варіанті індивідуального завдання. Визначте області допустимих значень параметрів формул (у разі непопадання значення x до ОДЗ – видавати відповідне повідомлення). Діапазон і крок зміни аргументу користувач задає в командному рядку, в тому разі, якщо в командному рядку параметри не задано або їх задано в недостатній кількості, програма пропонує здійснити ввід з клавіатури під час виконання програми. Передбачити перевірку правильності введених користувачем аргументів. Значення $F(x)$ обчислюють в окремій функції.

2. Складіть програму, яка обчислює значення функції, заданого за допомогою ряду Тейлора. Для ряду, члени якого обчислюються за формулою, що відповідає вашому індивідуальному завданню, підрахуйте та виведіть суму перших 5 членів ряду, суму членів ряду із уведеною точністю ϵ , кількість ітерацій та значення функції, що розрахована за допомогою стандартних математичних функцій. Організуйте вихід із програми за бажанням користувача. Обчислення значення функції із уведеною точністю ϵ , здійснити за допомогою трьох операторів циклу.

Варіант 11

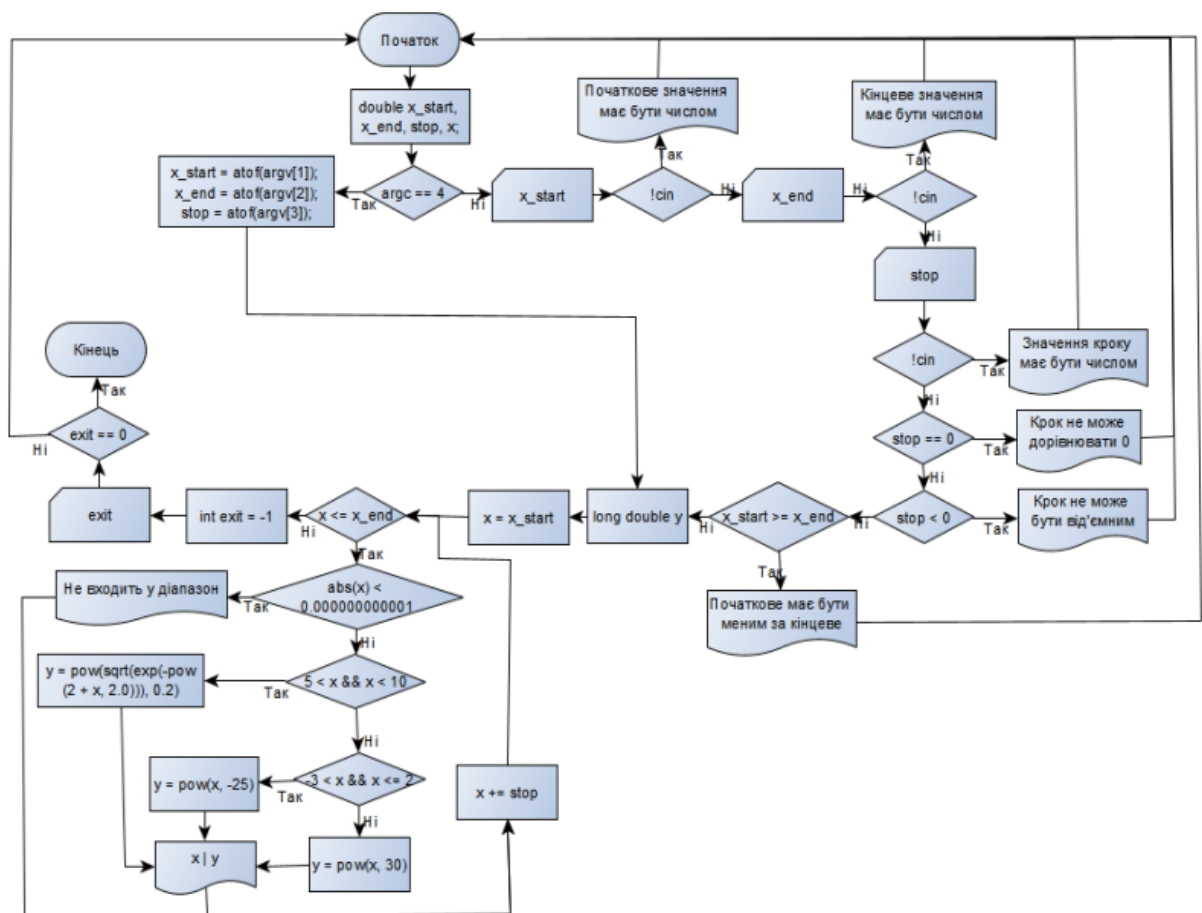
Завдання 1.

$$F(x) = \begin{cases} \sqrt[5]{e^{-2+x}} \frac{1}{\sqrt{x^2 + x^4 + \ln|x-3,14|}}, & 5 < x < 10; \\ x^{-25}, & -3 < x \leq 2; \\ x^{30}, & \text{в іншому випадку.} \end{cases}$$

Завдання 2. $\operatorname{arth}(x) = \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots, \quad |x| > 1.$

2. Опис розв'язку + блок-схема

Перше завдання:

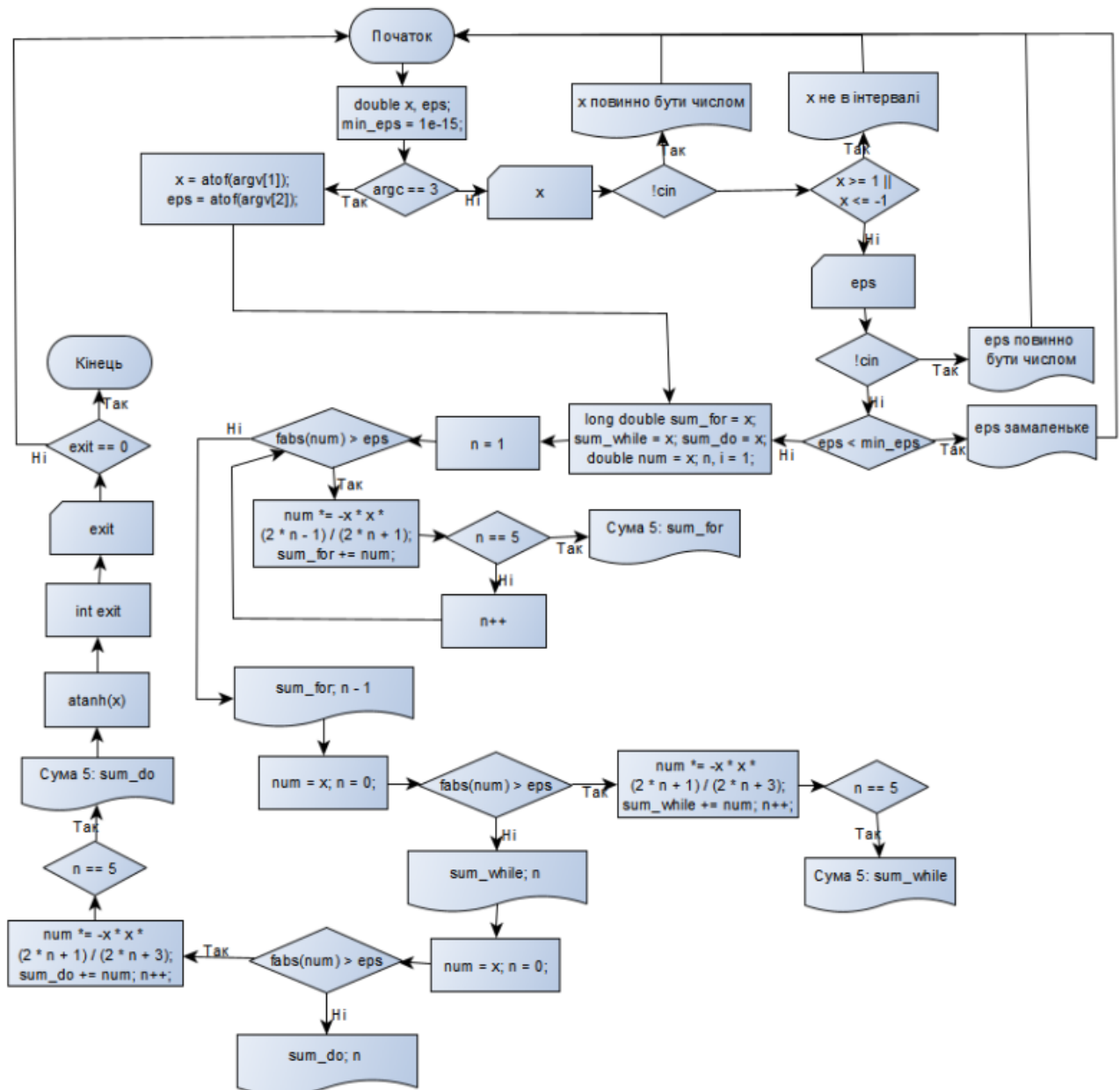


Програма виконує наступні дії:

- 1) встановлює кодову сторінку вводу та виводу консолі на 1251 для коректної роботи з кириличними символами;
- 2) програма дозволяє користувачеві ввести початкове та кінцеве значення інтервалу (x_start та x_end відповідно) та крок ($stop$), з яким будуть обчислюватися значення функції. Ці значення можна також передати як аргументи командного рядка при запуску програми. Якщо введені дані не є числами, або не відповідають логічним умовам (наприклад, крок дорівнює нулю або від'ємний), програма просить повторно ввести дані;
- 3) програма обчислює значення функції y для кожного x в інтервалі від x_start до x_end з кроком $stop$. Функція визначається за наступними правилами:
 - якщо x дуже близьке до нуля ($abs(x) < 0.00000000000001$), програма виводить повідомлення, що x не входить у діапазон;
 - якщо $5 < x < 10$, то $y = pow(sqrt(exp(-pow(2 + x, 2.0))), 0.2)$;
 - якщо $-3 < x \leq 2$, то $y = pow(x, -25)$;
 - у всіх інших випадках $y = pow(x, 30)$;
- 4) для кожного x програма виводить значення x та відповідне значення y у вигляді таблиці;
- 5) після виведення результатів програма пропонує користувачеві продовжити роботу або вийти з програми, вводячи 0.

Програма виконується в нескінченному циклі, що дозволяє користувачеві багаторазово вводити дані та отримувати результати без необхідності перезапуску програми.

Друге завдання:



Ця програма призначена для обчислення значення функції арка тангенс гіперболічний $\operatorname{atanh}(x)$ за допомогою ряду Тейлора. Програма виконує наступні дії:

- 1) встановлення кодової сторінки вводу та виводу консолі на 1251 для підтримки кирилических символів;
- 2) користувачу пропонується ввести значення x та точність обчислень ϵ . Перевіряється, що x належить інтервалу $(-1, 1)$ та ϵ не є занадто малим (більше за \min_eps);

3) значення функції $\operatorname{atanh}(x)$ обчислюється за допомогою ряду Тейлора. Сума ряду обчислюється тричі за допомогою різних циклів: `for`, `while` та `do while`. Для кожного циклу виводиться сума перших 5 членів ряду, сума всіх членів ряду (з урахуванням заданої точності `eps`), та кількість ітерацій, необхідних для досягнення заданої точності;

4) після обчислень програма виводить значення $\operatorname{atanh}(x)$ за допомогою вбудованої функції `atanh`, а також результати обчислення суми ряду Тейлора за допомогою різних циклів;

5) користувачу пропонується вибір: продовжити роботу програми з новими даними або завершити її.

Ця програма демонструє використання рядів Тейлора для апроксимації функцій та ілюструє роботу з різними циклами в мові програмування C++.

3. Вихідний текст програми розв'язку задачі (основні фрагменти з коментарями)

```
#include <iostream>
```

```
#include <iomanip>
```

```
#include <math.h>
```

```
#include <cstdlib>
```

```
using namespace std;
```

```
void text()
```

```
{
```

```
    cin.clear();
```

```
    cin.ignore(32767, '\n');
```

```
}
```

```
int main(int argc, char** argv)
```

```
{
```

```
while (true)
```

```
{
```

```
double x_start, x_end, stop, x;
```

```
if (argc == 4)
```

```
{
```

```
x_start = atof(argv[1]);
```

```
x_end = atof(argv[2]);
```

```
stop = atof(argv[3]);
```

```
}
```

```
else
```

```
{
```

```
while (true)

{

cout << "Введіть початкове значення інтервалу :";

cin >> x_start;

if (!cin)

{

cout << "Початкове значення має бути числом " << endl;

text();

continue;

}

cout << "Введіть кінцеве значення :";

cin >> x_end;

if (!cin)

{
```

```
cout << "Кінцеве значення має бути числом" << endl;
```

```
text();
```

```
continue;
```

```
}
```

```
cout << "Введіть значення кроку :";
```

```
cin >> stop;
```

```
if (!cin)
```

```
{
```

```
cout << "Значення кроку має бути числом" << endl;
```

```
text();
```

```
continue;
```

```
}
```

```
if (stop == 0)
```



```
{

cout << "Крок не може дорівнювати 0, введіть додатне число" << endl;

text();

continue;

}

else if (stop < 0)

{

cout << "Крок не може бути від'ємним, введіть додатне число" << endl;

text();

continue;

}

else if (x_start >= x_end)

{
```

```
cout << "Початкове значення повинно бути строго меншим за кінцеве" << endl;
```

```
text();
```

```
continue;
```

```
}
```

```
else break;
```

```
}
```

```
}
```

```
long double y;
```

```
cout << "_____ " << endl;
```

```
cout << setw(10) << "x" << setw(10) << "|" << setw(10) << "y" << endl;
```

```
for (x = x_start; x <= x_end; x += stop)
```

```
{
```

```
if (abs(x) < 0.000000000000001) {
```

```
cout << "_____ " << endl;
```

```
cout << setw(10) << "0" << setw(10) << "|" << setw(10) << "не входить у  
діапазон" << endl;
```

```
continue;
```

```
}
```

```
else if (5 < x && x < 10)
```

```
{
```

```
y = pow(sqrt(exp(-pow(2 + x, 2.0))), 0.2);
```

```
}
```

```
else if (-3 < x && x <= 2)
```

```
{
```

```
y = pow(x, -25);
```

```
}
```

```
else
```

```
{
```

```
y = pow(x, 30);
```

```
}
```

```
cout << "_____ " << endl;
```

```
cout << setw(10) << x << setw(10) << "|" << setw(15) << y << endl;
```

```
}int exit = -1;
```

```
cout << "Якщо бажаєте продовжити натисніть будь-яку цифру, якщо  
хочете вийти натисніть 0 :";
```

```
cin >> exit;
```

```
if (exit == 0)
```

```
{
```

```
break;
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

2 завд.

```
#include <iostream>
#include <math.h>
#include <iomanip>
#include <cstdlib>
using namespace std;

void words()
{
    cin.clear();
    cin.ignore(32767, '\n');
}

int main(int argc, char** argv)
{

    double x, eps;
    const double min_eps = 1e-15;
    while (true)
    {
        if (argc == 3)
        {
            x = atof(argv[1]);
            eps = atof(argv[2]);
        }
        else
```

```
{
while (true)
{
    cout << "X у вашому інтервалі = ";
    cin >> x;
    cout << endl;
    if (!cin)
    {
        cout << "x повинно бути числом, а не символом" << endl;
        words();
        continue;
    }
    if (x >= 1 || x <= -1) // Виправлено умову
    {
        cout << "x не в інтервалі";
        words();
        continue;
    }
    cout << "Уведіть eps =";
    cin >> eps;
    cout << endl;
    if (!cin)
    {
        cout << "eps повинно бути числом, а не символом" << endl;
        words();
        continue;
    }
    if (eps < min_eps)
```

```

    {
        cout << "eps замаленьке";
        words();
        continue;
    }
    break;
}
}

```

```

long double sum_for = x;
long double sum_while = x;
long double sum_do = x;
double num = x;
double n, i = 1;

```

```

for (n = 1; fabs(num) > eps; n++)
{
    num *= -x * x * (2 * n - 1) / (2 * n + 1);
    sum_for += num;
    if (n == 5)
        cout << "Сума перших 5 членів ряду Тейлора за допомогою цикла
for: " << setprecision(25) << sum_for << endl;
}

cout << "Сума всіх членів ряду Тейлора за допомогою цикла for: " <<
setprecision(25) << sum_for << endl;

cout << "Кількість ітерацій за допомогою цикла for: " << n - 1 << endl;

num = x;
n = 0;

```

```

while (fabs(num) > eps)
{
    num *= -x * x * (2 * n + 1) / (2 * n + 3);
    sum_while += num;
    n++;
    if (n == 5)
        cout << "Сума перших 5 членів ряду Тейлора за допомогою цикла
while: " << setprecision(25) << sum_while << endl;
}

cout << "Сума всіх членів ряду Тейлора за допомогою цикла while: "
<< setprecision(25) << sum_while << endl;

cout << "Кількість ітерацій за допомогою цикла while: " << n << endl;


num = x;
n = 0;
do
{
    num *= -x * x * (2 * n + 1) / (2 * n + 3);
    sum_do += num;
    n++;
    if (n == 5)
        cout << "Сума перших 5 членів ряду Тейлора за допомогою цикла
do while: " << setprecision(25) << sum_do << endl;
} while (fabs(num) > eps);

cout << "Сума всіх членів ряду Тейлора за допомогою цикла do while:
" << setprecision(25) << sum_do << endl;

cout << "Кількість ітерацій за допомогою цикла do while: " << n <<
endl;


cout << "\narth(x)=" << atan(x) << endl;

```



```

    int exit;

    cout << "Якщо бажаєте продовжити, натисніть будь-яку цифру. Для
виходу натисніть 0: ";

    cin >> exit;

    if (exit == 0)

        break;

}

return 0;

}

```

4. Опис інтерфейсу програми (керівництво користувача)

Інтерфейс першої програми:

- 1)запустіть .exe файл програми;
- 2)програма запропонує вам ввести початкове та кінцеве значення інтервалу, а також крок;
- 3)якщо ви введете нечислові символи, програма попередить вас і дозволить ввести дані знову;
- 4)якщо крок від'ємний або дорівнює нулю, програма також вас попередить і дозволить ввести дані знову;
- 5)якщо початкове значення більше або дорівнює кінцевому, програма попередить вас і дозволить ввести дані знову;
- 6)після введення коректних даних програма виведе таблицю з обрахованими значеннями x та y ;
- 7)програма запропонує завершити роботу, натиснувши 0, або продовжити з новими даними.

Інтерфейс другої програми:

- 1)запустіть .exe файл програми;
- 2)програма запропонує вам ввести значення x в інтервалі $|x| > 1$ та точність обчислень ϵ ;
- 3)якщо ви введете нечислові символи, програма попередить вас і дозволить ввести дані знову;

4)якщо значення x не входить в інтервал $(-1, 1)$, програма попередить вас і дозволить ввести дані знову;

5)якщо значення ϵ rs менше за мінімальне допустиме, програма попередить вас і дозволить ввести дані знову;

6)після введення коректних даних програма виведе суму перших п'яти членів ряду Тейлора та суму членів рядів з заданою точністю ϵ rs, обчислену за допомогою трьох різних циклів, кількість ітерацій для кожного циклу та порівняння з функцією $\operatorname{atanh}(x)$;

7)програма запропонує завершити роботу, натиснувши 0, або продовжити з новими даними.

5. Опис текстових прикладів

Введіть початкове значення інтервалу :2
Введіть кінцеве значення :8
Введіть значення кроку :4

| x | | y |
|---|--|-------------|
| 2 | | 2.98023e-08 |
| 6 | | 0.00166156 |

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :345

Введіть початкове значення інтервалу :d
Початкове значення має бути числом
Введіть початкове значення інтервалу :-1
Введіть кінцеве значення :4
Введіть значення кроку :2

| x | | y |
|----|--|-------------|
| -1 | | -1 |
| 1 | | 1 |
| 3 | | 2.05891e+14 |

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :3

Введіть початкове значення інтервалу :4
Введіть кінцеве значення :10
Введіть значення кроку :2

| x | | y |
|----|--|-------------|
| 4 | | 1.15292e+18 |
| 6 | | 0.00166156 |
| 8 | | 4.53999e-05 |
| 10 | | 1e+30 |

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :0

...Program finished with exit code 0
Press ENTER to exit console.

Введіть початкове значення інтервалу :1

Введіть кінцеве значення :-9

Введіть значення кроку :2

Початкове значення повинно бути строго меншим за кінцеве

Введіть початкове значення інтервалу :4

Введіть кінцеве значення :16

Введіть значення кроку :8

| x | | y |
|---|--|-------------|
| 4 | | 1.15292e+18 |

| | | |
|----|--|-------------|
| 12 | | 2.37376e+32 |
|----|--|-------------|

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :2

Введіть початков значення інтервалу :-5

Введіть кінцеве значення :15

Введіть значення кроку :10

| x | | y |
|----|--|-------------|
| -5 | | 9.31323e+20 |

| | | |
|---|--|-------------|
| 5 | | 9.31323e+20 |
|---|--|-------------|

| | | |
|----|--|-------------|
| 15 | | 1.91751e+35 |
|----|--|-------------|

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :0

...Program finished with exit code 0

Press ENTER to exit console.

```

Введіть початкове значення інтервалу :2
Введіть кінцеве значення :1
Введіть значення кроку :4
Початкове значення повинно бути строго меншим за кінцеве
Введіть початкове значення інтервалу :d
Початкове значення має бути числом
Введіть початкове значення інтервалу :4
Введіть кінцеве значення :6
Введіть значення кроку :-5
Крок не може бути від'ємним, введіть додатне число
Введіть початкове значення інтервалу :5
Введіть кінцеве значення :12
Введіть значення кроку :d
Значення кроку має бути числом
Введіть початкове значення інтервалу :1
Введіть кінцеве значення :18
Введіть значення кроку :3

```

| x | | y |
|----|--|-------------|
| 1 | | 1 |
| 4 | | 1.15292e+18 |
| 7 | | 0.000303539 |
| 10 | | 1e+30 |
| 13 | | 2.62e+33 |
| 16 | | 1.32923e+36 |

Якщо бажаєте продовжити натисніть будь-яку цифру, якщо хочете вийти натисніть 0 :0

```

...Program finished with exit code 0
Press ENTER to exit console.

```

X у вашому інтервалі = 0.2

Уведіть eps =0.5

```

Сума всіх членів ряду Тейлора за допомогою цикла for: 0.2000000000000000111022302
Кількість ітерацій за допомогою цикла for: 0
Сума всіх членів ряду Тейлора за допомогою цикла while: 0.2000000000000000111022302
Кількість ітерацій за допомогою цикла while: 0
Сума всіх членів ряду Тейлора за допомогою цикла do while: 0.197333333333333343657251
Кількість ітерацій за допомогою цикла do while: 1

```

arth(x)=0.1973955598498807750829087

Якщо бажаєте продовжити натисніть будь-яку цифру. Для виходу натисніть 0:

```
X у вашому інтервалі = 0.75
```

```
Уведіть eps =0.1
```

```
Сума всіх членів ряду Тейлора за допомогою цикла for: 0.6568359375000000013877788
```

```
Кількість ітерацій за допомогою цикла for: 2
```

```
Сума всіх членів ряду Тейлора за допомогою цикла while: 0.6568359375000000013877788
```

```
Кількість ітерацій за допомогою цикла while: 2
```

```
Сума всіх членів ряду Тейлора за допомогою цикла do while: 0.6568359375000000013877788
```

```
Кількість ітерацій за допомогою цикла do while: 2
```

```
arth(x)=0.6435011087932843709680242
```

```
Якщо бажаєте продовжити, натисніть будь-яку цифру. Для виходу натисніть 0: 76543
```

```
X у вашому інтервалі = 0.345
```

```
Уведіть eps =0.00000001
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла for: 0.3322134823336539974152851
```

```
Сума всіх членів ряду Тейлора за допомогою цикла for: 0.3322135500077206677054398
```

```
Кількість ітерацій за допомогою цикла for: 7
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла while: 0.3322134823336539974152851
```

```
Сума всіх членів ряду Тейлора за допомогою цикла while: 0.3322135500077206677054398
```

```
Кількість ітерацій за допомогою цикла while: 7
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла do while: 0.3322134823336539974152851
```

```
Сума всіх членів ряду Тейлора за допомогою цикла do while: 0.3322135500077206677054398
```

```
Кількість ітерацій за допомогою цикла do while: 7
```

```
arth(x)=0.3322135507465966974827154
```

```
Якщо бажаєте продовжити, натисніть будь-яку цифру. Для виходу натисніть 0: 5
```

```
X у вашому інтервалі = g
```

```
x повинно бути числом, а не символом
```

```
X у вашому інтервалі = 0.8
```

```
Уведіть eps =0.000000001
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла for: 0.6720140684892352131094007
```

```
Сума всіх членів ряду Тейлора за допомогою цикла for: 0.6747409419475277522195919
```

```
Кількість ітерацій за допомогою цикла for: 37
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла while: 0.6720140684892352131094007
```

```
Сума всіх членів ряду Тейлора за допомогою цикла while: 0.6747409419475277522195919
```

```
Кількість ітерацій за допомогою цикла while: 37
```

```
Сума перших 5 членів ряду Тейлора за допомогою цикла do while: 0.6720140684892352131094007
```

```
Сума всіх членів ряду Тейлора за допомогою цикла do while: 0.6747409419475277522195919
```

```
Кількість ітерацій за допомогою цикла do while: 37
```

```
arth(x)=0.6747409422235527420852463
```

```
Якщо бажаєте продовжити, натисніть будь-яку цифру. Для виходу натисніть 0: 0
```

```
...Program finished with exit code 0
```

```
Press ENTER to exit console.
```

6. Аналіз помилок (опис усунення зауважень)

У процесі виконання завдань виникли деякі помилки, зокрема, які стосуються не виконання всіх умов завдання. Для відповідності змінам у коді було необхідно переробити також блок-схему алгоритму та опис.

7. Висновки

Завдяки аналізу умов та розробці відповідних алгоритмів, успішно вирішила поставлені задачі. Також навчилася задавати та обробляти параметри `argc` та `argv`, що дозволяє запускати програму з різними вхідними даними без необхідності їх введення вручну під час виконання, навчилася перетворювати рядкові аргументи в числові значення, покращила здатність виявляти та виправляти помилки в програмі, забезпечуючи її коректну роботу за різних умов. Ця лабораторна робота допомогла поглибити розуміння основ програмування на мові C++, зокрема використання циклів, умовних операторів, обробки вхідних даних та роботи з командним рядком.