**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ**

**ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

**Индивидуальное задание №1**

**«Ряды Тейлора»**

**Вариант 11**

**по предмету «Алгоритмизация и программирование»**

**Выполнил: студент гр. 5130904/30002 Севостьянова А.В.**

**Руководитель Череповский Д.К.**

**Санкт-Петербург**

**2023 г.**

Оглавление

[1. Математические основы 3](#_Toc148734849)

[2. Общая постановка задачи 4](#_Toc148734850)

[3. Детальные требования и тест план 4](#_Toc148734851)

[a. Начало диапазона (start) должно быть задано корректно: 4](#_Toc148734852)

[b. Конец диапазона (end) должен быть задано корректно: 4](#_Toc148734853)

[c. Шаг диапазона (step) должен быть задан корректно: 4](#_Toc148734854)

[d. Точность вычислений (precision) должна быть задана корректно: 4](#_Toc148734855)

[Таблица с детальными требованиями и тест планом 5](#_Toc148734856)

[4. Программа 10](#_Toc148734857)

[main.cpp 10](#_Toc148734858)

[functions.h 11](#_Toc148734859)

[functions.cpp 11](#_Toc148734860)

[5. Таблица вычисленных значений функции 13](#_Toc148734861)

[Приложение A 14](#_Toc148734862)

[Приложение B 15](#_Toc148734863)

[Приложение C 17](#_Toc148734864)

[Приложение D 18](#_Toc148734865)

[Вывод 20](#_Toc148734866)

1. **Математические основы**

*Ряд Тейлора* - разложение функции в бесконечную сумму степенных функций.

**,** где *f(n)* – n-ая производная некой функции *f(х)*

Расписав данную формулу через сумму производных, получим бесконечный ряд слагаемых:

Будем использовать ряд Тейлора для вычисления значений функции *ex*.

Получаем следующее:

1. **Общая постановка задачи**

Написать функцию(expTaylor) для вычисления значения *ex* в заданной точке (x) с указанной абсолютной погрешностью вычисления (absError) при помощи ряда Тейлора. Написать программу вывода таблицы вычисленных значений функции на указанном(start, end) интервале и значений, полученных с использованием стандартных функций C++.

***Входные данные:***

* начало исследуемого диапазона(start)
* конец исследуемого диапазона(end)
* шаг интервала(step)
* точность вычислений(precision)

***Выходные данные:***

Таблица, содержащая вычисленное с помощью функции expTaylor и функции из стандартной библиотеки С++ значение для каждого х из заданного интервала.

1. **Детальные требования и тест план**
2. *Начало диапазона (start) должно быть задано корректно:*

* start – дробное число
* |start| < 1

1. *Конец диапазона (end) должен быть задано корректно:*

* end – дробное число
* |end| < 1
* start <= end

1. *Шаг диапазона (step) должен быть задан корректно:*

* step – дробное число
* step > 0.0

1. *Точность вычислений (precision) должна быть задана корректно:*

* precision – целое число
* precision – положительное число
* 0 < precision <= 12

## *Таблица с детальными требованиями и тест планом*

|  |  |  |  |
| --- | --- | --- | --- |
| Требования | Детальные требования | Данные | Ожидаемый результат |
| 1. ***Начало диапазона должно быть задано корректно(см. приложение A)*** | | | |
| 1. start – число | Если start принимает нечисловое значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | R | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 2. start – дробное число | Если start принимает целочисленное значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | 0 | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 3. |start| < 1 | Если  |start| >= 1 , то сообщение: «Границы диапазона указаны неверно», завершение работы программы. | 7.0 | Сообщение: «Границы диапазона указаны неверно»  Завершение работы программы. |
| 4. start – задано корректно | Если start задано корректно продолжение работы программы. | 0.4 | Продолжение работы программы. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. ***Конец диапазона должно быть задано корректно(см. приложение B)*** | | | |
| 1. end – число | Если end принимает нечисловое значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | R | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 2. end – дробное число | Если end принимает целочисленное значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | 0 | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 3. |end| < 1 | Если |end| >= 1 , то сообщение: «Границы диапазона указаны неверно», завершение работы программы. | 7.0 | Сообщение: «Границы диапазона указаны неверно»  Завершение работы программы. |
| 4. end >= start | Если end <start, то сообщение: «Границы диапазона указаны неверно» | 0.3  (start = 0.4 по пункту а) | Сообщение: «Границы диапазона указаны неверно»  Завершение работы программы. |
| 5. end – задан корректно | Если end задан корректно продолжение работы программы. | 0.7 | Продолжение работы программы. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. ***Шаг диапазона должен быть задано корректно(см. приложение C)*** | | | |
| 1. step – число | Если step принимает нечисловое значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | R | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 2. step – дробное число | Если step принимает целочисленное значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | 0 | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 3. step >0 | Если step <= 0 , то сообщение: «Неверный шаг диапазона», завершение работы программы. | 0.0  -7.0 | Сообщение: «Неверный шаг диапазона»  Завершение работы программы. |
| 4. step – задан корректно | Если step задан корректно, то продолжение работы программы. | 0.02 | Продолжение работы программы. |

|  |  |  |  |
| --- | --- | --- | --- |
| 1. ***Точность вычислений должна быть задана корректно(см. приложение D)*** | | | |
| 1. precision – число | Если precision принимает нечисловое значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | R | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 2. precision – целое число | Если precision принимает дробное значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | 0.5 | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 3. precision - неотрицательное число | Если precision принимает отрицательное значение, то сообщение: «Некорректный ввод данных», завершение работы программы. | -7 | Сообщение: «Некорректный ввод данных»  Завершение работы программы. |
| 4. 0< precision <= 12 | Если  precision > 12 или  precision == 0, то сообщение:  «Некорректная точность вычислений», завершение работы программы. | 19 | Сообщение:  «Некорректная точность вычислений»  Завершение работы программы. |
| 5. precision – задана корректно. | Если precision задана корректно, то продолжение работы программы. | 7 | Продолжение работы программы.  Вызов функции expTaylor, подсчет значений *ex* для каждого числа из заданного диапазона. Вывод таблицы, содержащей результаты функции expTaylor и стандартной функции С++ |

# 4. Программа

## *main.cpp*

#include <iostream>

#include<string>

#include<exception>

#include <iomanip>

#include "functions.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

string str = ""; // хранение переменных до проверки их корректности

double start = 0.0, // начало введенного пользователем диапазона

end = 0.0, // конец введенного пользователем диапазона

step = 0.0; // шаг интервала

int precision = 0; // точность вычислений

try

{

cout << "Введите начало диапазона\n x1 =\t ";

cin >> str;

start = checkDouble(str);

cout << "Введите конец диапазона\n x2 =\t ";

cin >> str;

end = checkDouble(str);

if (abs(start) >= 1 || abs(end) >= 1 || start> end) throw runtime\_error("Границы диапазона указаны неверно");

cout << "Введите шаг диапазона\n step =\t ";

cin >> str;

step = checkDouble(str);

if (step <= 0.0) throw runtime\_error("Неверный шаг диапазона");

cout << "Введите точность вычислений\n precision =\t ";

cin >> str;

precision = checkUnsigntInt(str);

if (precision > 12) throw runtime\_error("Слишком большая точность вычислений");

}

catch (runtime\_error &ex)

{

cerr << ex.what() << "\n";

exit(EXIT\_FAILURE);

}

double absError = pow(0.1, precision); // абсолютная погрешность

cout << " \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout <<"||" << setw(4) << "x" << setw(4) << "|" << setw(14) << "expTaylor" << setw(5) << "|";

cout << setw(10) << "exp" << setw(10) << "||" << endl;

cout << "||\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_||\n";

for (start; start < end; start += step)

{

if (abs(start) < 1e-6) start = 0.0;

cout << "||" << setw(5) << start << setw(3) << "|" << setprecision(precision +1) << setw(16) << expTaylor(start, absError) << setw(3) << "|"

<< setw(16) << exp(start)<<setw(4)<<"||" << endl;

}

cout << "||" << setw(5) << end << setw(3) << "|" << setprecision(precision + 1) << setw(16) << expTaylor(end, absError) << setw(3) << "|"

<< setw(16) << exp(end) << setw(4) << "||" << endl;

cout << "||\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_||\n";

return 0;

}

## *functions.h*

#pragma once

#include <iostream>

#include <string>

#include <regex>

using namespace std;

double checkDouble(string &x); // x - переменная, вводимая пользователем

/\*

Функция проверяет корректность пользовательского ввода.

Введенное число (x) может принимать дробное значение, причем

|x| < 1

\*/

int checkUnsigntInt(const string& x); // x - переменная, вводимая пользователем

/\*

Функция проверяет корректность пользовательского ввода.

Введенное число (x) может принимать исключительно целое значение

\*/

double expTaylor(const double x, const double absError); // x - точка, для которой выполняется расчет;

// absError - абсолютная погрешность, показатель прекращения вычислений

/\*

Функция вычисляющая e^x при помощи ряда Тейлора

e^x = 1 + x/1 + x^2/2! + x^3/3! + x^4/4! .....

\*/

## *functions.cpp*

#include "functions.h"

double checkDouble(string &x)

{

regex regular("-?\\d+,\\d+");

regex replace("\\.");

x = regex\_replace(x, replace, ",");

if (regex\_match(x, regular)) return stod(x);

else throw runtime\_error("Некорректный ввод данных");

}

int checkUnsigntInt(const string &x)

{

regex regular ("\\d+");

if (regex\_match(x,regular) && stoi(x) != 0) return stoi(x);

else throw runtime\_error("Некорректный ввод данных");

}

double expTaylor(const double x, const double absError)

{

double xInPow = 1.0;

double res = 1.0;

int factorial = 1;

for (int i = 1; ; i++)

{

xInPow \*= x;

factorial \*= i;

res += xInPow / factorial;

if (abs(xInPow / factorial) <= absError) break;

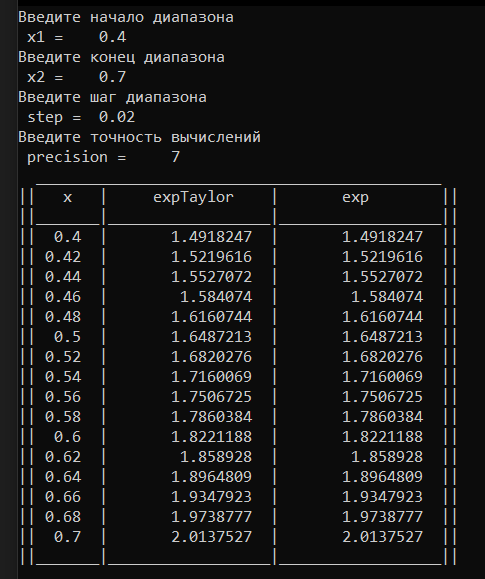
}

return res;

}

# 5. Таблица вычисленных значений функции

Построение таблицы для х ∈ [0,4: 0,7]

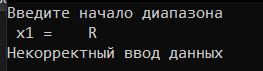


*Рис. 1 Таблица вычисленных значений функции*

# Приложение A

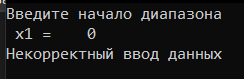
Начало диапазона должно быть задано корректно:

1. *start – число*



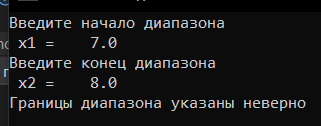
*Рис. 1 start принимает нечисловое значение*

1. *start – имеет дробное значение*



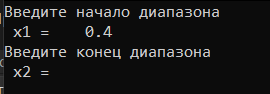
*Рис. 2 start принимает целочисленное значение*

1. *|start| < 1*



*Рис. 3 start по модулю превышает 1*

1. *start задано корректно*

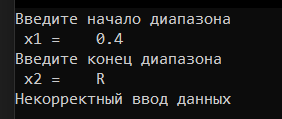


*Рис. 4 start задан корректно*

# Приложение B

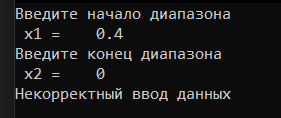
Конец диапазона должен быть задан корректно:

1. *end – число*



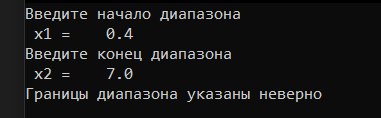
*Рис. 1 end принимает нечисловое значение*

1. *end – имеет дробное значение*



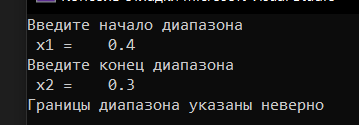
*Рис. 2 end принимает целочисленное значение*

1. *|end| < 1*



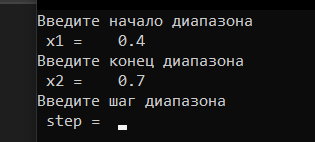
*Рис. 3 end по модулю превышает 1*

1. *end >= start*



*Рис. 4 end превышает start*

1. *end задан корректно*

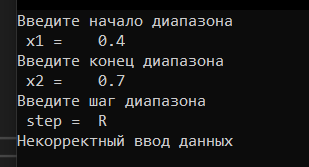


*Рис. 5 end задан корректно*

# Приложение C

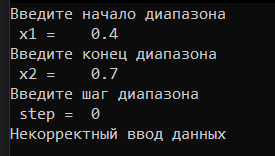
Шаг диапазона должен быть задан корректно:

1. *step – число*



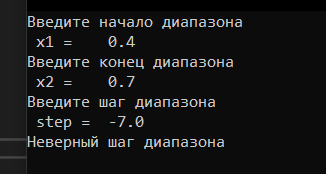
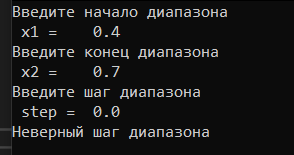
*Рис. 1 step принимает нечисловое значение*

1. *step – имеет дробное значение*



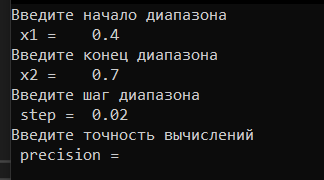
*Рис. 2 step принимает целочисленное значение*

1. *step >* 0



*Рис. 3 step принимает отрицательное значение*

1. *step задан корректно*

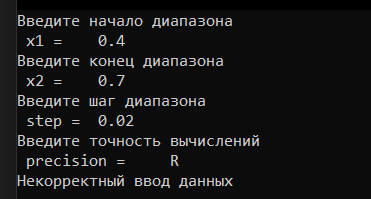


*Рис. 4 step задан корректно*

# Приложение D

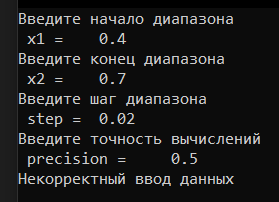
Точность вычислений должна быть задана корректно:

1. *precision – число*



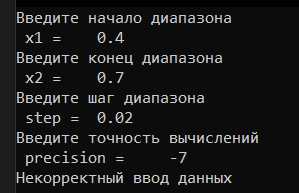
*Рис. 1 precision принимает нечисловое значение*

1. *precision – имеет целочисленное значение*



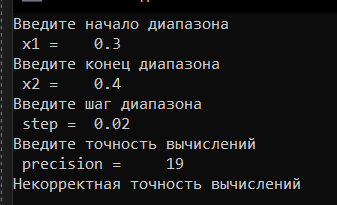
*Рис. 2 precision принимает дробное значение*

1. *precision – неотрицательное число*



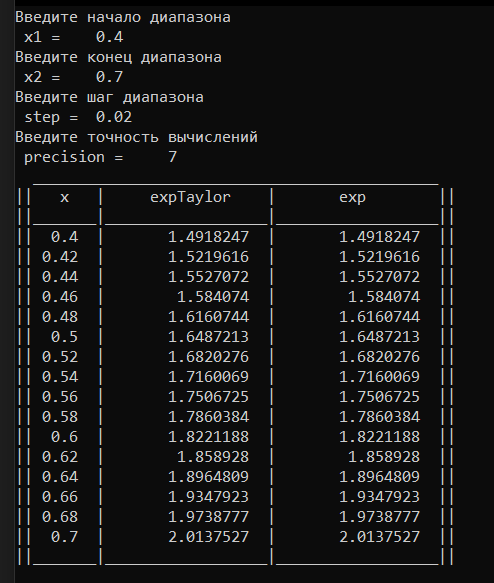
*Рис. 3 precisionпринимает отрицательное значение*

1. *0< precision <=12*



*Рис. 4 precision равно «0» или превышает 12*

1. *precision задана корректно*



*Рис. 5 precision задана корректно*

**Вывод**

В ходе работы над программой:

1. Были изучены ряды Тейлора, и на их основе была написана функция, вычисляющая значение *ех.*
2. Выполняемый код был разделен на логические части - функции, что облегчает его восприятие.
3. Был использован оператор try … catch для более эффективной обработки ошибок.
4. Были использованы исключения, что также повышает эффективность и удобство обработки возникающих ошибок.
5. Для проверки ввода были использованы регулярные выражения.
6. Была решена проблема отображения нуля в типе double (например,

-1.0408341e-17)