Тоббо Кирилл Вадимович

Группа: 3530904/10005

Лабораторная работа №2. Индивидуальные задания.

**Задание 1.10**

**Общая постановка задачи.**

Напишите функцию (в соответствии с указанным вариантом) для вычисления значения предложенной суммы в заданной точке (x) с указанной абсолютной погрешностью вычисления (absError) и максимальным числом слагаемых (numberMax). Вычислить сумму в соответствие с формулой: cosh(x) = 1 + + + , если x (-1;1)

**Детальные требования, тест план.**

1. Значение x должно быть задано корректно. Если x меньше -1 или больше 1, выводится сообщение: “x must be greater than 1 and less than -1 and x must be a number. End of the program.”
2. Значение step должно быть задано корректно. Если step задан не числом или step > x, выводится сообщение: “step must be less than x and must be a number. End of the program.”
3. Значение numberMax должно быть задано корректно. Если numberMax меньше 1, выводится сообщение: “The maximum number of elements must be greater than 1 or equal to 1.”
4. Значение absError должно быть задано корректно. Если absError задано не числом, выводится сообщение: “Absolute error must be a number.”
5. Создаем цикл, который будет выводит интервал значений x.
6. Создаем функцию(calcPow), отвечающую за подсчет степени числа.
7. Создаем функцию(calcCosH), отвечающую за подсчет суммы элементов.
8. Если один из элементов окажется больше абсолютной погрешности, возвращаем значение -3.
9. В функции main запускаем функцию calcCosH. Если данная функция возвращает значение -3, выводим сообщение: “Absolute error can’t be reach.” Иначе возвращаем значение суммы.

**Код:**

double calcPow(double in, int power);

double calcCosH(double x, int termMaxNumb, double absError);

double myRound(double in, double absError);

int main()

{

double finish = 0.0;

int termMaxNumb = 1;

double absError = 1.0;

double start = 0.0;

double step = 0.0;

std::cout << "Enter right boundary of the interval, left boundary of the interval, step, max number of terms and absolute error:";

try

{

std::cin >> start;

if (std::cin.fail() || start > 1.0 || start < -1.0 || (std::cin.peek() != 10 && std::cin.peek() != 32))

{

throw ERROR\_INVALID\_FIRST\_BORDER\_INPUT;

}

std::cin >> finish;

if (std::cin.fail() || finish > 1.0 || finish < -1.0 || (std::cin.peek() != 10 && std::cin.peek() != 32) || finish < start)

{

throw ERROR\_INVALID\_SECOND\_BORDER\_INPUT;

}

std::cin >> step;

if (std::cin.fail() || (std::cin.peek() != 10 && std::cin.peek() != 32) || termMaxNumb < 1)

{

throw ERROR\_INVALID\_STEP\_INPUT;

}

std::cin >> termMaxNumb;

if (std::cin.fail() || (std::cin.peek() != 10 && std::cin.peek() != 32) || termMaxNumb < 1)

{

throw ERROR\_INVALID\_TERM\_MAX\_NUMB\_INPUT;

}

std::cin >> absError;

if (std::cin.fail() || std::cin.peek() != 10)

{

throw ERROR\_INVALID\_ABS\_ERROR\_INPUT;

}

for (double i = start; i < finish; i += step) {

double result = calcCosH(i, termMaxNumb, absError);

if (result == -3)

{

throw ERROR\_CANT\_REACH\_ABS\_ERROR;

}

std::cout << "x = " << " " << myRound(i, step) << "\t," << "result: " << myRound(result, absError) << std::endl;

}

}

catch (const char\* error)

{

system("cls");

std::cerr << std::endl << error << std::endl;

return -1;

}

}

double calcCosH(double x, int termMaxNumb, double absError)

{

int i = 1;

double term = 0.0;

double sum = 1.0;

double lastTerm = 0.0;

do

{

term = calcPow(x, i \* 2) / calcFact(i \* 2);

sum += term;

i++;

lastTerm = term;

} while (i <= termMaxNumb and abs(term - lastTerm) < absError);

if (termMaxNumb == i)

{

return -3;

}

return sum;

}

double calcPow(double in, int power)

{

double res = in;

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

{

res \*= in;

}

return res;

}

double calcFact(double in)

{

double res = 1;

for (int i = 2; i < in + 1; i++)

{

res \*= i;

}

return res;

}

double myRound(double in, double absError)

{

return floor(in \* (1 / absError)) / (1 / absError);

}