**Постановка задачи**

Разработать детальные требования, тест план и программу для следующей задачи:

Дана функция:, абсолютная погрешность вычисления (absError) и максимальным числом слагаемых. Нужно написать программу вывода таблицы вычисленных значений функции на указанном интервале (-1;1) и значений, полученных с использованием стандартных функций С++.

**Задача 1. Функции**

1. Дана функция: , абсолютная погрешность вычисления (absError) и максимальным числом слагаемых. Нужно написать программу вывода таблицы вычисленных значений функции на указанном интервале (-1;1) и значений, полученных с использованием стандартных функций С++.

**Входные данные:** Точность вычисления, максимальное число слагаемых, интервал, на котором проводятся вычисления, шаг интервала.

**Выходные данные:** Таблица вычисленных значений функции на указанном интервале и значений, полученных с использованием стандартных функций C++.

1. **Пример**:

Точность погрешности:0.1;

Максимальное число слагаемых:2;

интервал: (-1;1);

шаг интервала: 0.1;

Посчитаем корень, оставляя в формуле только 2 первых слагаемых: f (0.1) = 1 + ½\*0.1= 1.05. Измерим точность погрешности. Следующее слагаемое по модулю было бы равно 1+½\*0.1 – 1/(2\*4)\*(0.1)2 = 1.04625. Так как оно меньше absError, то указанная точность достигается и мы выводим значение полученного корня. Аналогично считаем и для других х.

1. **Детальные требования:**
   1. Вводимые данные должны быть заданы корректно.
      1. Вводимые данные – числа.
         1. Если хотя бы одно из вводимых данных не число, то вывести сообщение «You went out of range or entered not a number, program termination!»
      2. Для вводимых величин должны выполняться следующие условия -1 <= x1 <= 1, -1 <= x2 <= 1, -1 < absError < 1, count > 0, где absError – абсолютная погрешность измерения, count – максимальное число слагаемых, step – шаг интервала, x1 – левая граница интервала, x2 – правая граница интервала.
         1. Если хотя бы одно условие для вводимых величин не выполняется, то вывести сообщение: «You went out of range or entered not a number, program termination!»
      3. Указанная точность должна достигаться.
         1. Если указанная точность не достигается, то вывести сообщение: «The required accuracy has not been achieved!»
   2. Программа должна выдавать требуемые значения, если введенные данные корректны и расчет величин возможен.
      1. Введенные данные корректны и расчет величин возможен.
         1. Если введенные данные корректны и расчет величин возможен, то вывести таблицу вычисленных значений функции на указанном интервале и значений, полученных с использованием стандартных функций C++..

**Тест-план:**

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Детальные требования | Данные | Ожидаемый результат |
| 3.1. Вводимые данные должны быть заданы корректно. | | | |
| 3.1.1 Вводимые данные – числа | 3.1.1.1 Если хотя бы одно из вводимых данных не число, то вывести cообщение: «You went out of range or entered not a number, program termination!» | x | Сообщение: «You went out of range or entered not a number, program termination!» |
| 3.1.2 Для вводимых величин должны выполняться следующие условия -1 <= x1 <= 1, -1 <= x2 <= 1, -1 < absError < 1, count > 0, где absError – абсолютная погрешность измерения, count – максимальное число слагаемых, step – шаг интервала, x1 – левая граница интервала, x2 – правая граница интервала. | 3.1.2.1 Если хотя бы одно условие для вводимых величин не выполняется, то вывести сообщение: «You went out of range or entered not a number, program termination!» | -1 1  2  2  0.1 | Cообщение: «You went out of range or entered not a number, program termination!» |
| 3.1.3. Указанная точность должна достигаться | * + - 1. Если указанная точность не достигается, то вывести сообщение: «The required accuracy has not been achieved!» | -1 1  2  0.1  0.1 | Для х, где точность не достигается, вывод строки с сообщением: « The required accuracy has not been achieved!».  The required accuracy has not been achieved!  The required accuracy has not been achieved!  The required accuracy has not been achieved!  The required accuracy has not been achieved!  x = -0.5; my result = 0.8; functiion result = 0.7  x = -0.4; my result = 0.8; functiion result = 0.8  x = -0.3; my result = 0.8; functiion result = 0.8  x = -0.2; my result = 0.9; functiion result = 0.9  x = -0.1; my result = 0.9; functiion result = 0.9  x = -0.0; my result = 1.0; functiion result = 1.0  x = 0.1; my result = 1.0; functiion result = 1.0  x = 0.2; my result = 1.1; functiion result = 1.1  x = 0.3; my result = 1.1; functiion result = 1.1  x = 0.4; my result = 1.2; functiion result = 1.2  x = 0.5; my result = 1.2; functiion result = 1.2  The required accuracy has not been achieved!  The required accuracy has not been achieved!  The required accuracy has not been achieved!  The required accuracy has not been achieved! |
| 3.2. Программа должна выдавать требуемые значения, если введенные данные корректны и расчет величин возможен | | | |
| 3.2.1 Введенные данные корректны и расчет величин возможен | 3.2.1.1 Если введенные данные корректны и расчет величин возможен, то вывести таблицу вычисленных значений функции на указанном интервале и значений, полученных с использованием стандартных функций C++. | -0.5 0.5  2  0.1  0.1 | x = -0.5; my result = 0.8; functiion result = 0.7  x = -0.4; my result = 0.8; functiion result = 0.8  x = -0.3; my result = 0.8; functiion result = 0.8  x = -0.2; my result = 0.9; functiion result = 0.9  x = -0.1; my result = 0.9; functiion result = 0.9  x = -0.0; my result = 1.0; functiion result = 1.0  x = 0.1; my result = 1.1; functiion result = 1.0  x = 0.2; my result = 1.1; functiion result = 1.1  x = 0.3; my result = 1.1; functiion result = 1.1  x = 0.4; my result = 1.2; functiion result = 1.2  x = 0.5; my result = 1.2; functiion result = 1.2 |

**Код:**

#include <iostream>

#include <math.h>

#include <exception>

#include <iomanip>

const char\* ERROR\_INPUT\_FAIL = "You went out of range or entered not a number, program termination!";

const char\* ERROR\_INPUT\_X\_LOW\_HI = "The entered x is out of range, program termination!";

double funcZ(int a);

double funcH(int a);

double funci(double a, int count);

double mypow(double xF, int degree);

int main()

{

int N = 0, j = 0, count = 0, k = 0;

double x = 0.0, absError = 0.0, step = 0.0, result = 0.0, x1 = 0.0, x2 = 0.0, absError1 = 0.0, rounding = 1.0;

try

{

std::cout << "Enter x1 and x2: ";

std::cin >> x1 >> x2;;

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

{

throw ERROR\_INPUT\_FAIL;

}

if ((x1 > 1) or (x1 < -1) or (x2 > 1) or (x2 < -1))

{

throw ERROR\_INPUT\_X\_LOW\_HI;

}

std::cout << "Enter number of digits: ";

std::cin >> count;

if ((std::cin.fail()) or (std::cin.peek() != 10) or (count <= 0))

{

throw ERROR\_INPUT\_FAIL;

}

std::cout << "Enter the absolute error: ";

std::cin >> absError;

if ((std::cin.fail()) or (std::cin.peek() != 10) or abs(absError) >= 1)

{

throw ERROR\_INPUT\_FAIL;

}

absError = abs(absError);

absError1 = absError;

while (absError1 < 1)

{

absError1 \*= 10;

k++;

}

std::cout << "Enter spacing step: ";

std::cin >> step;

if ((std::cin.fail()) or (std::cin.peek() != 10) or (step < 0))

{

throw ERROR\_INPUT\_FAIL;

}

double i = 0;

if (x1 == -1)

{

i = x1 + step;

}

else

{

i = x1;

}

if (x2 == 1)

{

x2 = x2 - abs(step);

}

for (i; i <= x2; i += abs(step))

{

result = funci(i, count);

if (abs(funci(i, count + 1) - funci(i, count)) < absError)

{

std::cout << std::fixed << std::setprecision(k) << "x = " << i << "; my result = " << result << "; functiion result = " << sqrt(1 + i) << std::endl;

}

else

{

std::cout << "The required accuracy has not been achieved!" << std::endl;

}

}

}

catch (const char\* error)

{

std::cerr << error;

return -1;

}

return 0;

}

double mypow(double xF, int degree)

{

int i = 0;

double function = 1.0;

for (i = 1; i <= degree; i++)

{

function \*= xF;

}

return function;

}

double funcH(int a)

{

int mean = 1;

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

{

if (i % 2 != 0)

{

mean \*= i;

}

}

return mean;

}

double funcZ(int a)

{

int i, mean = 1;

if (a != 1)

{

for (i = 1; i < a; i++)

{

mean \*= 2 \* i;

}

}

else

{

return 1;

}

return mean;

}

double funci(double a, int count)

{

int k = 0;

double f = 1.0;

for (int i = 2; i <= count; i++)

{

if (i % 2 == 0)

{

k = 1;

}

else

{

k = -1;

}

f = f + k \* double(funcH(i) / funcZ(i)) \* mypow(a, i - 1);

}

return f;

}