Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Севастопольский государственный университет»

Институт радиоэлектроники и информационной безопасности

Кафедра «Радиоэлектроника и телекоммуникации»

**ОТЧЁТ**

по лабораторной работе №1

«Программная реализация рекурсивных функций»

по дисциплине

«Программирование»

Выполнил: студент гр. РС/с-19-1-о

Охрименко Д. А.

Вариант: 18

Защитил с оценкой: \_\_\_\_\_\_

Принял: ст. пр. Дурманов М.А.

Севастополь

2020

# Содержание

1. Описание лабораторной работы 3

1.1. Цель работы 3

1.2. Задача и текст индивидуального задания 3

1.3. Теоретические сведения 3

2. Выполнение лабораторной работы 4

2.1. Структурная схема алгоритма работы программы 5

2.2. Текст программы 6

2.3. Таблица переменных 7

2.4. Результаты выполнения программы 8

Выводы 9

# 1.описание лабораторной работы

## Цель работы

Изучение особенностей составления и использования рекурсивных функций в языках программирования *C*/*C*++.

Получение практических навыков реализации алгоритмов, использующих рекурсивные функции, средствами языков программирования *C*/*C*++.

## Текст индивидуального задания

Составить схему алгоритма и написать на языке *С*/*С*++ программу вычисления суммы *n* элементов ряда, полученного при разложении функции *f*(*x*), используя рекурсивные функции. Варианты задания выбираются по указанию преподавателя из таблицы 1.1. Значения аргумента *x* выбираются произвольно. Результаты вычисления суммы элементов ряда вывести на экран в формате с плавающей точкой и сравнить со значением функции *f*(*x*).





Таблица 1.1 — Исходные данные для выполнения задания № 1



## 1.3. Теоретические сведения

**1.3.1.** **Рекурсивные** **функции**

Рекурсивной называют функцию, которая вызывает саму себя. Такая рекурсия называется прямой. Существует косвенная рекурсия, когда две или более функций вызывают друг друга.

Примером рекурсивной функции является вычисление факториала. Для того чтобы получить значение факториала числа *n*, требуется умножить на *n* факториал числа *n* 1:

**long** **factorial** **(long** **n)** **{**

**if** **(n==0** **||** **n==1)** **return** **1;**

**return** **(n** **\*** **factorial(n** **-** **1));** **}**

// 0! = 1, 1! = 1,

// нерекурсивная ветвь функции // рекурсивный вызов функции

В функции выполняется проверка значения аргумента и если аргумент равен **1**или **0**, то функция возвращает значение **1**:

**if** **(n==0** **||** **n==1)** **return** **1;**

Данная ветвь функции не является рекурсивной и предназначена для выхода из рекурсии. Рекурсивный вызов функции выполняется до тех пор, пока аргумент не равен **0**или **1**.

Рекурсивный вызов функции вычисления факториала записан в строке

**return** **(n** **\*** **factorial (n** **-** **1));**

Здесь выполняется умножение числа *n* на факториал числа *n* 1, т.е. вызывается функция вычисления факториала от декрементрированного на единицу аргумента и полученное значение факториала умножается на значение аргумента.

Рекурсивную функцию чаще всего применяют для компактной реализации рекурсивных алгоритмов, а также для работы со структурами данных, описанных рекурсивно. Любую рекурсивную функцию можно реализовать без применения рекурсии, для чего необходимо в программе обеспечить хранение всех необходимых данных самостоятельно.

Достоинством использования рекурсии является компактность программного кода, а недостатком — повышенный расход ресурсов вычислительной системы и опасность переполнения стека.

# 2.Выполнение ЛАБОРАТОРНОЙ РАБОТЫ

## 2.1. Структурная схема алгоритма работы программы

## Структурная схема алгоритма решения задачи представлена на рис. 2.1.

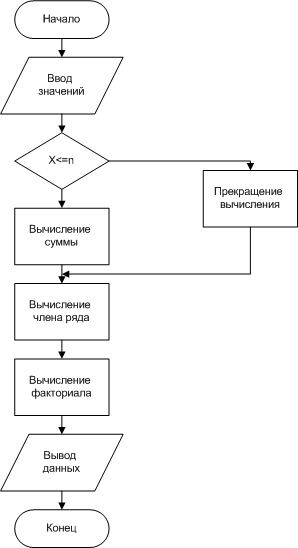
****

Рисунок 2.1 —Схема алгоритма решения задачи

**2.2. Текст программы**

#include<conio.h> // подключение библиотеки управления

// вводом-выводом средствами консоли MSDOS

#include<iostream> // подключение библиотеки стандартных

//средств ввода-вывода языка С++

#include<math.h> // подключение библиотеки математических функций

#include<locale> // подключение библиотеки локализации

// объявление прототипов функций

// функция вычисления суммы ряда

double sum\_of\_series(double x,int k,int n);

// функция вычисления члена ряда

double term\_of\_series(double x,int k);

// функция вычисления факториала

long int factorial (long int k);

const float pi = 3.141592654;

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus"); // локализация

int n; // количество суммируемых членов ряда

double x, // значение аргумента

sum; // сумма членов ряда

cout <<"Введите значение аргумента 'x':";

cin >>x;

cout<<"Введите количество элементов ряда 'n':";

cin>>n;

sum=(sum\_of\_series(x,1,n)); // вычисление суммы членов ряда

cout<<"Сумма элементов ряда : "<<sum<<endl;

cout << "Значение функции sin2(x)" << ((1-x)/x)\*log(1/1-x) << endl;

system("pause");

return(0);

}

// рекурсивная функция вычисления суммы ряда

double sum\_of\_series(double x, int k,int n)

{ // вычисление суммы

if (k<=n)

{

return(term\_of\_series(x,k)+sum\_of\_series(x,k+1,n));

}

else

{ // прекращаем вычисления суммы, если количество членов

// ряда превысило заданное количество

return 0;

}

}

// функция вычисления члена ряда

double term\_of\_series(double x,int k)

{

double result = 0;

result=pow(x,k)/(k\*(k+1));

cout <<"Элемент ряда №"<<k<<" : "<<result<<endl;

return result;

}

**2.3. Таблица переменных**

В программе использованы переменные, представленные в таблице 2.3.

Таблица 2.3 — Используемые переменные

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Тип** | **Значение** |
| *X* | *Double* | Значение аргумент |
| *N* | *int* | Количество суммируемых членов ряда |
| *Sum* | *Double* | Сумма членов ряда |
| *K* | *int* | Член ряда |

**2.4. Результаты выполнения программы**

Результаты выполнения программы представлены на рисунке 2.2. Результаты проверки работы программы представлены на рисунке 2.3.

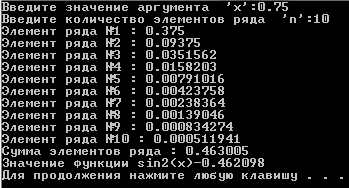
**

Рисунок 2.2 —Результат выполнения программы

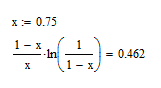


Рисунок 2.3 —Результаты вычисления функции в MathCAD

# ВЫВОДЫ

В ходе выполнения лабораторной работы были изучены особенности составления и использования рекурсивных функций в языках программирования C/C++.

Получены практические навыки реализации алгоритмов, использующих рекурсивные функции, средствами языков программирования C/C++.

После ручной проверки решения задачи и проверки в MathCAD, полученные результаты совпали с расчётами, что свидетельствует о правильности работы программы.