МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Учебный Центр Информационных Технологий «Информатика»

Лабораторная работа №3

по дисциплине «Информатика и программирование I часть»

Направление подготовки: 230105 - «Программное обеспечение вычислительной техники и автоматизированных систем»

Выполнил слушатель: Аветисян Арташес Робертович
Вариант: 4
Дата сдачи: 04.10.2020
Преподаватель: Прытков Д. В.

Новосибирск, 2020г.

1 Задание

Изучение итерационных процессов и циклов.

Для заданного варианта написать функцию вычисления суммы ряда. для диапазона значений 0.1 .. 0.9 и шага 0.1 изменения аргумента вычислить значения суммы ряда и контрольной функции, к которой он сходится, с точностью до 4 знаков после запятой.

Вариант 4:

$$1 + x\ln 3 + \frac{(x\ln 3)^2}{2!} + \ldots + \frac{(x\ln 3)^n}{n!} \qquad 3^x$$

2 Теоретический материал

Операторы цикла в Си

Оператор for

Оператор for – это наиболее общий способ организации цикла.

```
for ( выражение 1 ; выражение 2 ; выражение 3 ) тело
```

Выражение 1 обычно используется для установления начального значения переменных, управляющих циклом. Выражение 2 – это выражение, определяющее условие, при котором тело цикла будет выполняться. Выражение 3 определяет изменение переменных, управляющих циклом после каждого выполнения тела цикла.

Оператор while

Оператор цикла while называется циклом с предусловием. Его синтаксис:

```
while (выражение) тело ;
```

Выражение в скобках может принимать ненулевое (истинное) или нулевое (ложное) значение. Если оно истинно, то выполняется тело цикла и выражение вычисляется снова. Если выражение ложно, то цикл while заканчивается. Оператор while удобно использовать в ситуациях, когда тело оператора не всегда нужно выполнять.

Оператор do while

Основным отличием между циклами while и do – while является то, что тело в цикле do – while выполняется по крайней мере один раз. Тело цикла будет выполняться до тех пор, пока выражение в скобках не примет ложное значение. Если оно ложно при входе в цикл, то его тело выполняется ровно один раз. Его синтаксис:

```
do тело while (выражение);
```

Чтобы прервать выполнение цикла до того, как условие станет ложным, можно использовать оператор break.

Оператор continue

Оператор continue, как и оператор break, используется только внутри операторов цикла, но в отличие от него выполнение программы продолжается не с оператора, следующего за прерванным оператором, а с начала прерванного оператора.

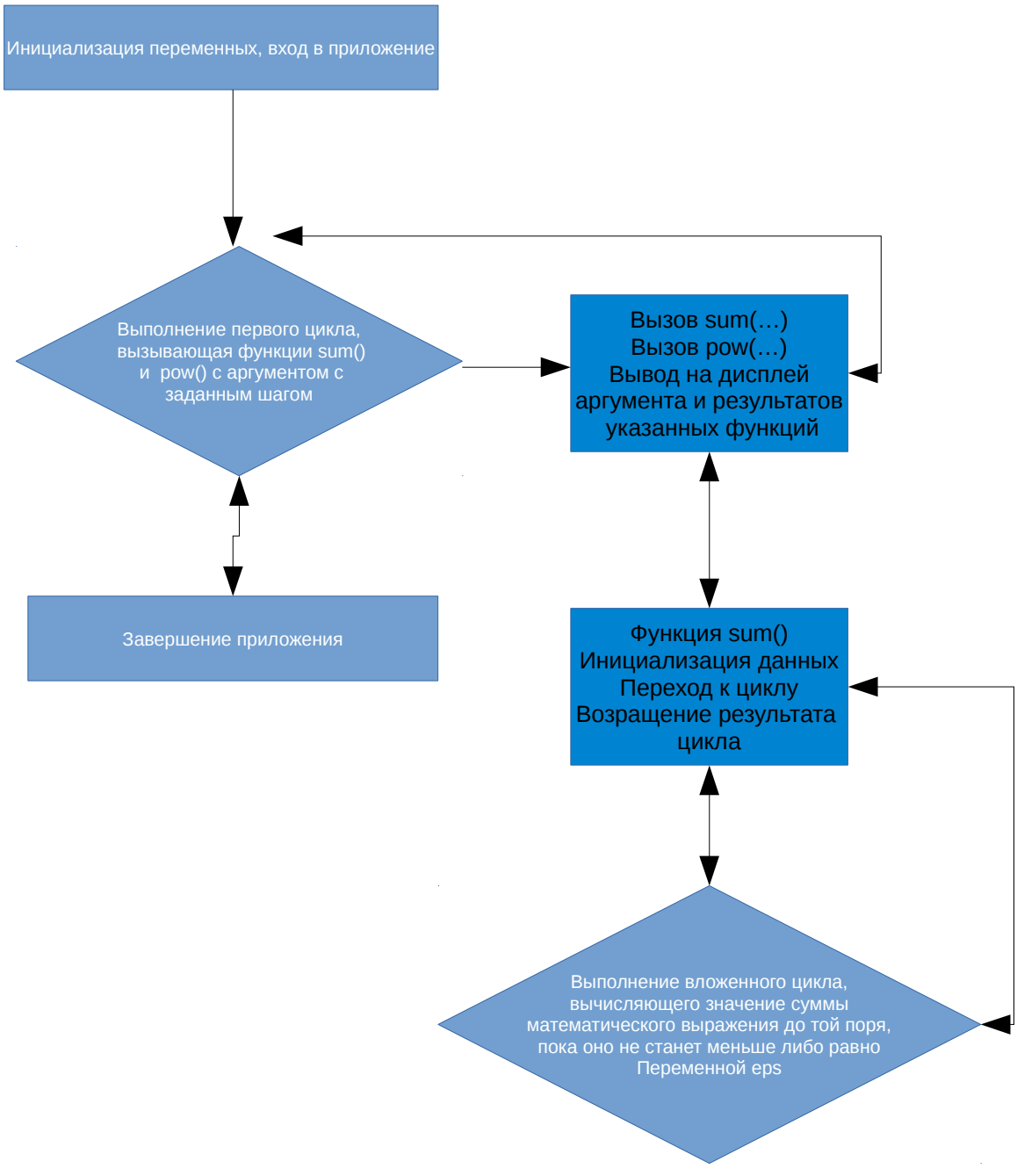
Оператор return

Оператор return завершает выполнение функции, в которой он задан, и возвращает управление в вызывающую функцию, в точку, непосредственно следующую за вызовом. Использование оператора return необходимо либо для немедленного выхода из функции, либо для передачи возвращаемого значения.

Оператор break

Оператор break обеспечивает прекращение выполнения самого внутреннего из объединяющих его операторов switch, do, for, while. После выполнения оператора break управление передается оператору, следующему за прерванным.

3 Описание алгоритма приложения



4 Описание реализации

Препроцессорные директивы:

#include - используется для включения в исходный код заголовочных файлов библиотек

Подключаемые заголовочные файлы библиотеки:

- stdio.h
- stdlib.h
- math.h

Используемые функции:

int main()

Указанная основная функция, являющаяся начальной точкой для выполнения программы. Она обычно управляет выполнением программы, вызывая другие ее функции. Как правило, выполнение программы завершается в конце функции **main**, но по разным причинам это может случиться и в других местах программы. Может быть ипользована для передачи аргументов в приложение.

int printf(const char *format, arg-list)

Функция printf() записывает в stdout аргументы из списка arg-list под управлением строки, на которую указывает аргумент format. Строка, на которую указывает format, состоит из объектов двух различных назначений. Во- первых, это символы, которые сами должны быть выведены на экран. Во-вторых, это спецификаторы формата, определяющие вид, в котором будут выведены аргументы из списка arg-list. Спецификаторы формата состоят из символа процент, за которым следует код формата.

double sum (double x, double eps)

Возвращает результат вычисления математического выражения (x ln3)^0 / 0! + (x ln3)^1 / 1! + (x ln3)^2 / 2! +... (x ln3)^n / n! с аргументом **double x,** с таким числом итерации, при котором результат суммы калькуляций меньше либо равен **double eps.**

long int factorial(int n)

Возвращает значение факториала **int n**

double pow(double x, double y)

Возвращает значение, равное x, возведенному в степень y

Используемые конструкции и операторы:

if

Оператор выбора **if** позволяет выполнять или опустить выполнять определенных участков кода, в зависимости от того является ли истинным или ложным условие этого оператора. Одно из самых важных назначений оператора выбора **if** так это то, что он позволяет программе совершить действие на выбор, в зависимости от того, например, какие данные ввел пользователь.

for

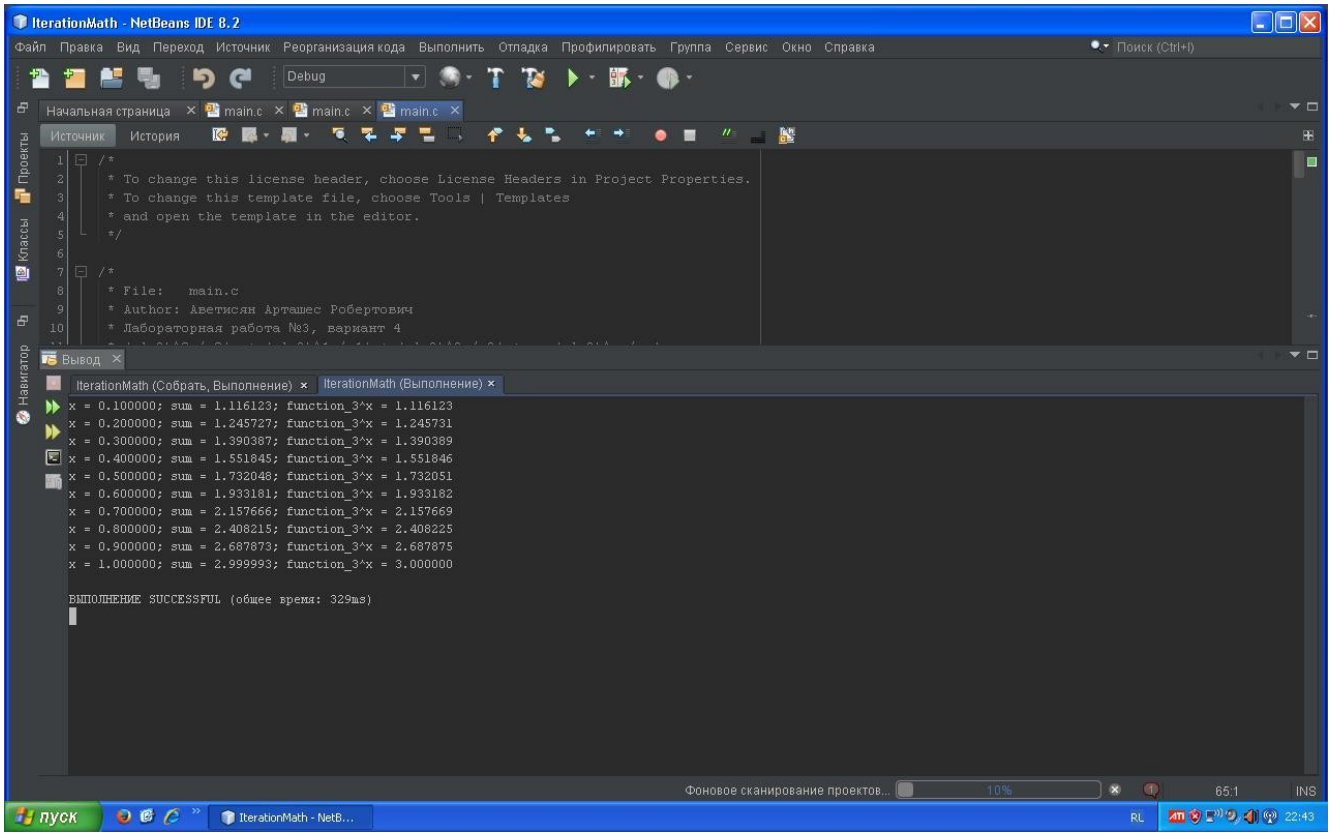
Параметрический цикл (цикл с заданным числом повторений), реализуемый при помощи операций инициализация (присваивание параметру цикла начального значения), проверки условия повторения цикла, модификации - изменения значения параметра для следующего прохождения тела цикла

5 Пример работы программы

На рисунке 1 показана фотография дисплея компьютера с развернутым окном среды разработки NetBeans IDE 8.2 в WindowsXP после выполнения разработанного приложения.

В результате работы приложения, построчно выведены значения аргементов, суммы в итерационном цикле с применением соответствующего аргумента, а также значение функции.

Рисунок 1. Фотография экрана во время выполнения приложения, разработанного согласно задачи лабораторной работы



Выводы

Изучен теоретическим материал по работе итерационными циклами в Си, который применен в практике для решения задачи в рамках данной лабораторной работы.

Успешно разработан и апробирован алгоритм вычисления суммы ряда и коэфициента.

Приложение. Текст программы с комментариями

```
/*
 * File:   main.c
 * Author: Аветисян Арташес Робертович
 * Лабораторная работа №3, вариант 4
 * (xln3)^0 / 0!  + (xln3)^1 / 1! + (xln3)^2 / 2! +... (xln3)^n / n!
 *
 * Created on 12 сентября 2020 г., 22:34
 */

#include <stdio.h>
#include <stdlib.h>
```

```
#include <math.h>

long int factorial(int n);
double sum (double x, double eps);

int main(int argc, char** argv) {
    double x;
    for (x = 0.1; x <= 1; x += 0.1){

        printf("x = %lf; sum = %lf; function_3^x = %lf\n", x, sum(x, 0.0001), pow(3, x));
    }

    return (EXIT_SUCCESS);
}

double sum (double x, double eps){

    //xln3)^0 / 0!  + (xln3)^1 / 1! + (xln3)^2 / 2! +... (xln3)^n / n!

    double overallSumm = 0.;
    double currentIterationResult = x;
    double logResult = log(3);
    double logResulMulX = logResult * x;

    for (int n = 0; fabs(currentIterationResult) > eps; n++){

        currentIterationResult = pow(logResulMulX, n) / factorial(n);
        overallSumm += currentIterationResult;
    }

    return overallSumm;
}

long int factorial(int n) {

    if (n >= 1){

        return n * factorial(n-1);
    } else {

        return 1;

    }
}
```

Защита: