ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
О ЛАБОРАТОРНОЙ РАБОТЕ №3-2

По дисциплине «Языки программирования»

ВАРИАНТ 1

Выполнил: ст. гр. ТКИ - 141

Абузина Елизавета Юрьевна

Проверил: к.т.н., доц. Васильева М. А.

(Проверил: к.т.н, доц. Балакина Е. П.)

Москва 2024

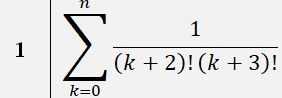
1. Формулировка задания

Составьте две программы:

1. вычислить сумму первых *n* членов последовательности (*k* = 1, 2, 3 ..., *n*).
2. вычислить сумму всех членов последовательности, не меньших заданного числа *e*.

Помните о проверке пользовательского ввода. Все результаты вывести на экран. Отчёт дополнить блок-схемой. При вычислении факториалов рекомендуется отказаться от использования рекурсивных методов.

Таблица  – Исходные данные



1. Блок-схема алгоритма

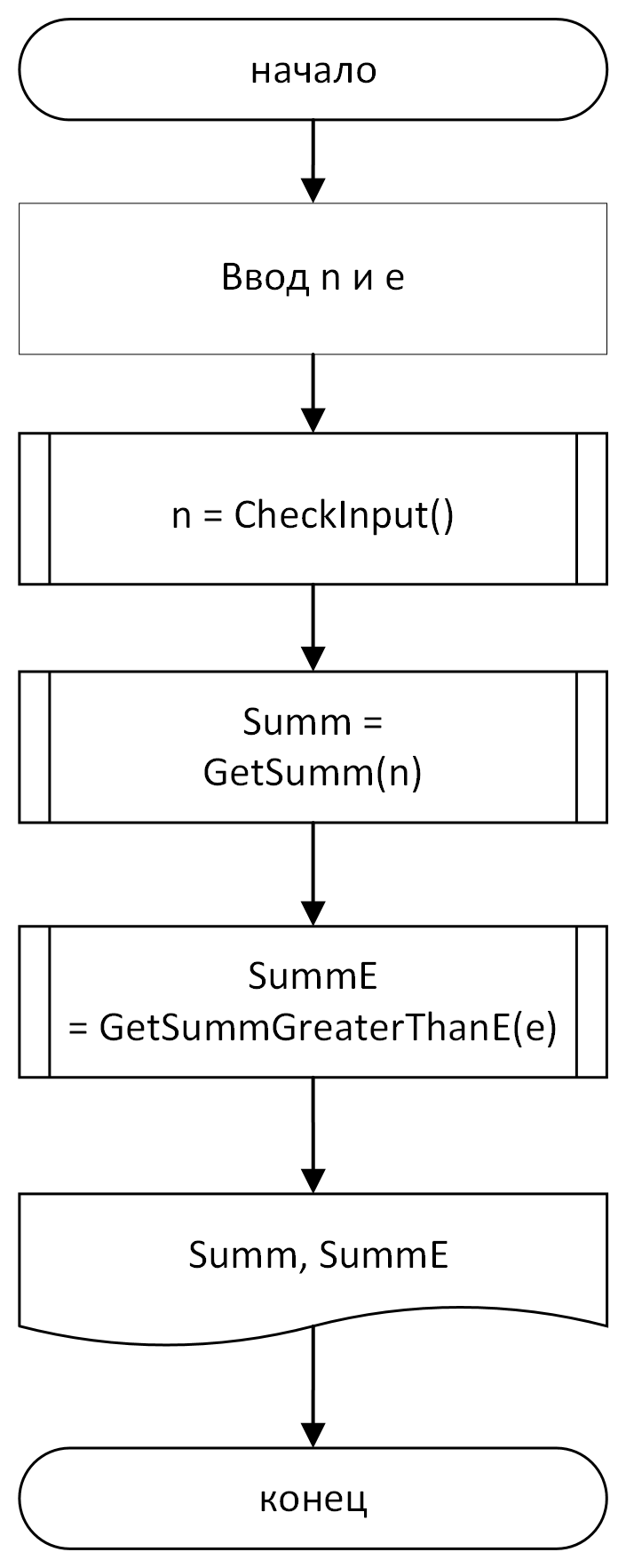


Рисунок  ­ Блок-схема основного алгоритма

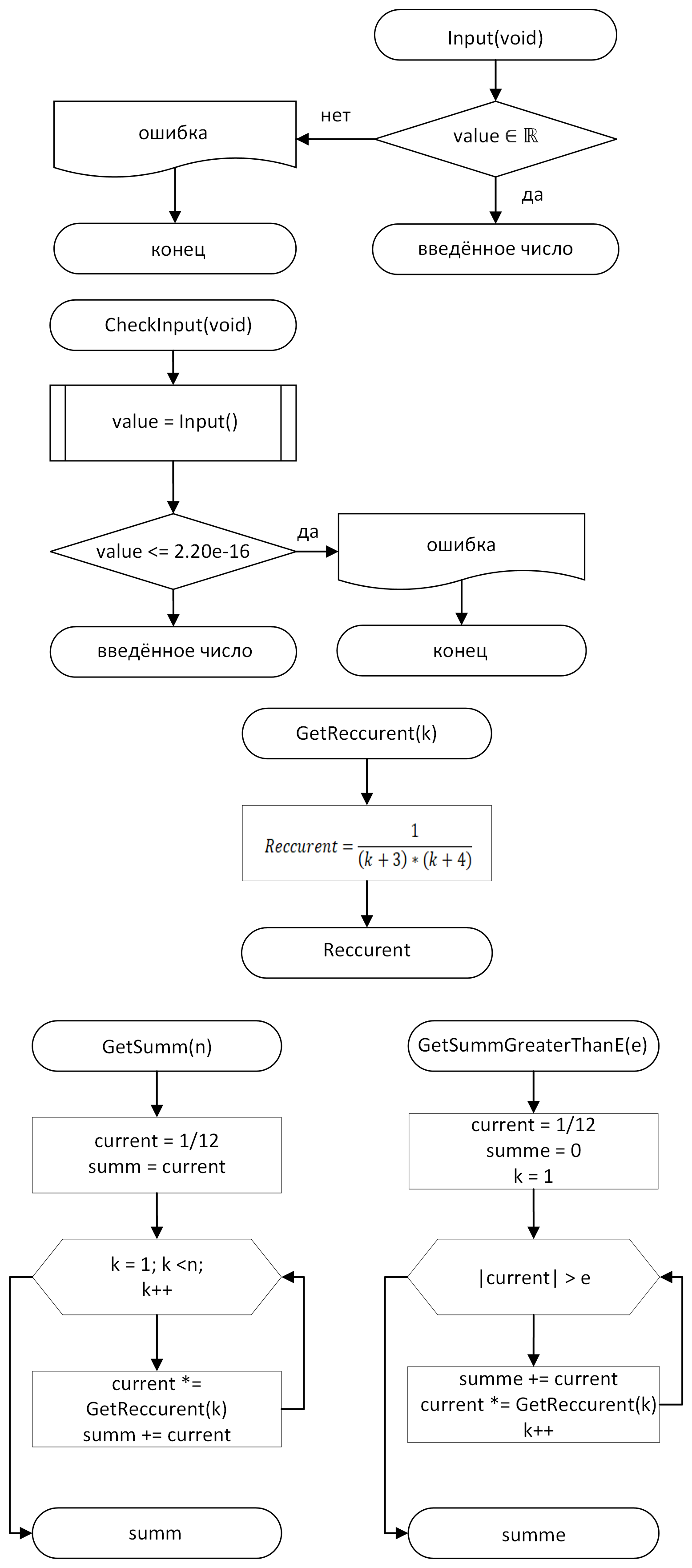


Рисунок  – Блок-схема используемых функций

1. Текст программы на языке C

#include <stdio.h>

#include <stdlib.h>

#include <errno.h>

#include <locale.h>

#include <float.h>

#include <math.h>

/\*\*

\* @brief Считывает вещественное число

\* @remarks При неправильном вводе программа завершает выполнение.

\* @return Вещественное число

\*/

double Input(void);

/\*\*

\* @brief Считывает вещественное число

\* @remarks При отрицательном значении ввода программа завершает выполнение

\* @return Вещественное число

\*/

double CheckInput(void);

/\*\*

\* @brief Получает на вход число k

\* @return Рекурентное число с заданным k

\*/

double GetReccurent(const double k);

/\*\*

\* @brief Считывает номер последнего члена последовательности

\* @return Сумма рекурентных чисел

\*/

double GetSumm(const double n);

/\*\*

\* @brief Вычисляет сумму членов последовательности, не меньших e

\* @param e Пороговое значение

\* @return Сумма членов последовательности, не меньших e

\*/

double GetSummGreaterThanE(const double e);

/\*\*

\* @brief Точка входа в программу

\* @return Код ошибки

\*/

int main(void)

{

setlocale(LC\_ALL, "Russian");

puts("Введите номер последнего члена последовательности");

double n = CheckInput();

printf("Сумма при n = %lf равна %.18lf\n", n, GetSumm(n));

puts("Введите пороговое число");

double e = CheckInput();

printf("Сумма членов последовательности, не меньших e = %lf равна %.18lf", e, GetSummGreaterThanE(e));

}

double Input(void)

{

double value;

int result = scanf("%lf", &value);

if (result != 1)

{

errno = EIO;

perror("Не удалось считать число");

exit(EXIT\_FAILURE);

}

return value;

}

double CheckInput(void)

{

double value = Input();

if (value <= DBL\_EPSILON)

{

errno = EDOM;

printf("Ошибка ввода числового значения");

exit(EXIT\_FAILURE);

}

return value;

}

double GetSumm(double n)

{

double current = 1.0 / 12;

double summ = current;

for (int k = 1; k < n; k++)

{

current \*= GetReccurent(k);

summ += current;

}

return summ;

}

double GetReccurent(double k)

{

return 1/((k + 3) \* (k + 4));

}

double GetSummGreaterThanE(double e)

{

double current = 1.0 / 12;

double summ = 0.0;

double k = 1.0;

while (fabs(current) >= e)

{

summ += current;

current \*= GetReccurent(k);

k++;

}

return summ;

}

1. Результаты выполнения программы

Результаты выполнения программы представлены ниже (Рисунок 3).

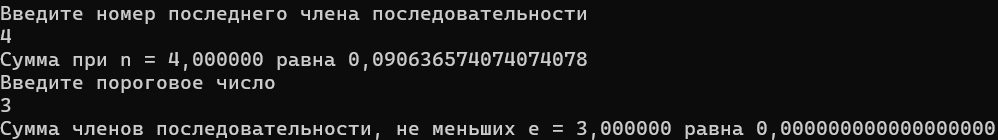


Рисунок  – Результаты выполнения программы

1. Выполнение тестовых примеров

В программе IDLE выполнен тестовый пример. Результаты его выполнения представлен ниже (Рисунок 4).

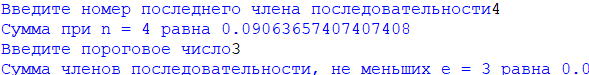


Рисунок  – Результат работы программы

1. Отметка о выполнении задания в веб-хостинге системы контроля версий

