КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по дисциплине

«Программирование в системах информационной безопасности»

на тему

«Регулярные и итерационные циклы»

Вариант 9

Выполнил:

студент группы БСУ1801

Копылов М.А.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# 1. Цель работы

Овладеть навыками создания циклических алгоритмов на языке C с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.

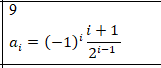
# Задание

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

Все указанные ниже функции разработать в трёх вариантах: с использованием цикла for, while, do … while. Функциям из задания дать одинаковые имена. Каждую функцию поместить в отдельный файл. Прототип функции вынести в заголовочный файл. В решении создать три проекта.

**Индивидуальное задание**

Вариант 9



**Общее Задание**

#### Задание 1

Написать функцию для вычисления выражения .

Функция в качестве параметра принимает значение и вычисляет сумму слагаемых.

Прототип функции double summ(int n).

Функция не должна использовать функции консольного ввода-вывода.

#### Задание 2

Написать функцию для вычисления выражения с точностью .

Функция в качестве параметра принимает значение и вычисляет сумму до тех пор, пока не выполнится условие .

Прототип функции double summ2(double eps).

Функция не должна использовать функции консольного ввода-вывода.

#### Задание 3

Написать функцию, которая печатает членов последовательности , исключая из неё каждый -ый член.

Числа и передаются в функцию в качестве параметра. Для исключения члена последовательности используйте инструкцию перехода continue.

Прототип функции void print(int n, int k).

Пример вывода: 1.2 1.3 0.75

#### Задание 4

Написать функцию, которая возвращает номер первого члена последовательности , для которого выполняется условие .

Функция в качестве параметра принимает значение .

Выход из цикла организовать с помощью инструкции .

Прототип функции int findFirstElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

#### Задание 5

Написать функцию, которая возвращает номер первого отрицательного члена последовательности , для которого выполняется условие .

Функция в качестве параметра принимает значение .

Выход из цикла организовать с помощью инструкции перехода return.

Прототип функции int findFirstNegativeElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

#### Задание 6

Разработать функцию main(), которая в цикле выводит на экран меню из 6 пунктов и ожидает ответа пользователя:

1. Задание 1
2. Задание 2
3. Задание 3
4. Задание 4
5. Задание 5
6. Выход

Обработку пользовательского ввода выполнить с помощью инструкции switch. При нажатии на соответствующую цифру меню стирается с экрана и приложение переходит в режим демонстрации работы выбранного задания. По завершении демонстрации программа ожидает пользовательского ввода для перехода в меню. Выход из программы должно осуществляться при нажатии цифры 6 в режиме меню. Скомпилировать 3 версии программы:

1. Все функции реализованы с помощью цикла for
2. Все функции реализованы с помощью цикла while
3. Все функции реализованы с помощью цикла do … while

Убедиться, что все три версии программы работают одинаково

**Выполнение**

Листинг 1 – Исходный код файла header.h

double summ(int n);

double summ2(double eps);

void print(int n, int k);

int findFirstElement(double eps);

int findFirstNegativeElement(double eps);

Листинг 2 – Исходный код файла task1.c (for)

#include <math.h>

double summ(int n)

{

double sum = 0;

for (int i = 0; i < n; i++)

{

sum += pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

}

return sum;

}

Листинг 3 – Исходный код файла task1.c (while)

#include <math.h>

double summ(int n)

{

double sum = 0;

int i = 0;

while (i < n)

{

sum += pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

++i;

}

return sum;

}

Листинг 4 – Исходный код файла task1.c (do while)

#include <math.h>

double summ(int n)

{

double sum = 0;

int i = 0;

if (n == 0)

{

return 0;

}

else

{

do

{

sum += pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

++i;

} while (i < n);

return sum;

}

}

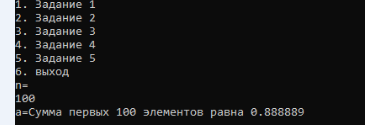


Рисунок 1 – Результат работы task1.c

Листинг 5 – Исходный код файла task2.c (for)

#include <math.h>

double summ2(double eps)

{

double sum = 0;

int i = 0;

double a;

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

for (i = 0; fabs(a) > eps; i++)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

sum += a;

}

return sum;

}

Листинг 6 – Исходный код файла task2.c (while)

#include <math.h>

double summ2(double eps)

{

double sum = 0;

int i = 0;

double a;

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

while (fabs(a) > eps)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

sum += a;

i++;

}

return sum;

}

Листинг 7 – Исходный код файла task2.c (do while)

#include <math.h>

double summ2(double eps)

{

double sum = 0;

int i = 0;

double a;

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps)

{

return a;

}

else

{

do

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

sum += a;

i++;

} while (fabs(a) > eps);

}

return sum;

}

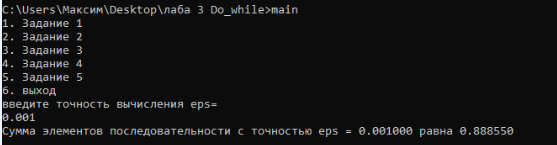


Рисунок 2 – Результат работы task2.c

Листинг 8 – Исходный код файла task3.c (for)

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double a = 0;

for (int i = 0; i < n; i++)

{

if ((i + 1) % k == 0)

{

continue;

}

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

printf("%lf ", a);

}

printf("\n");

}

Листинг 9 – Исходный код файла task3.c (while)

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double a = 0;

int i = 0;

while (i < n)

{

if ((i + 1) % k == 0)

{

++i;

continue;

}

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

printf("%lf ", a);

++i;

}

printf("\n");

}

Листинг 10 – Исходный код файла task3.c (do while)

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double a = 0;

int i = 0;

do

{

if ((i + 1) % k == 0)

{

++i;

continue;

}

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

printf("%lf ", a);

++i;

} while (i < n);

printf("\n");

}

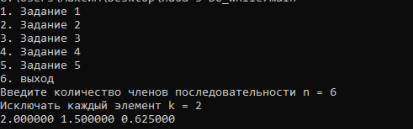


Рисунок 3 – Результат работы task3.c

Листинг 11 – Исходный код файла task4.c (for)

#include <math.h>

int findFirstElement(double eps)

{

int i = 0;

double a;

for (i; ;i++)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps)

{

break;

}

}

return ++i;

}

Листинг 12 – Исходный код файла task4.c (while)

#include <math.h>

int findFirstElement(double eps)

{

int i = 0;

double a;

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

while (fabs(a) > eps)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps)

{

break;

}

++i;

}

return ++i;

}

Листинг 13 – Исходный код файла task4.cpp

#include <math.h>

int findFirstElement(double eps)

{

int i = 0;

double a;

do

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps)

{

break;

}

++i;

} while (fabs(a) > eps);

return ++i;

}

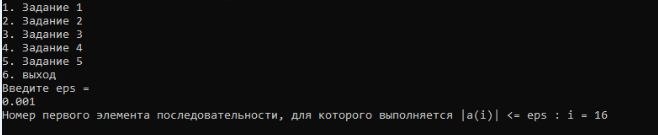


Рисунок 4 – Результат работы task4.c

Листинг 14 – Исходный код файла task5.c (for)

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i = 0;

double a;

for (i; ;i++)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps && a < 0)

{

return ++i;

}

}

}

Листинг 15 – Исходный код файла task5.c (while)

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i = 0;

double a;

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

while (fabs(a) > 0)

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps && a < 0)

{

return ++i;

}

++i;

}

}

Листинг 16 – Исходный код файла task5.c (do while)

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i = 0;

double a;

do

{

a = pow(-1, i) \* (i + 1) / (pow(2, (i - 1)));

if (fabs(a) <= eps && a < 0)

{

return ++i;

}

++i;

} while ((fabs(a) > 0));

}

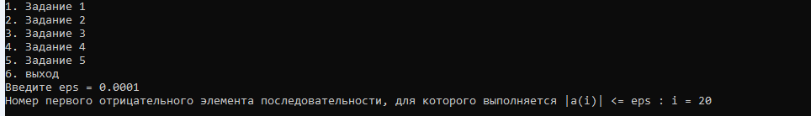


Рисунок 5 – Результат работы программ task5.c

Листинг 17 – Исходный код файла main.c

#include <locale.h>

#include <stdio.h>

#include "header.h"

int main(void) {

int n = 0;

int k = 0;

double sum = 0;

double eps = 0;

setlocale(LC\_CTYPE, "rus");

printf("1. Задание 1 \n2. Задание 2 \n3. Задание 3 \n4. Задание 4 \n5. Задание 5 \n6. выход \n");

switch (getch()) {

case '1':

printf("n=");

scanf\_s("%d", &n);

sum = summ(n);

printf("a=");

printf("Сумма первых %d элементов равна %lf \n", n, sum);

break;

case '2':

printf("введите точность вычисления eps=");

scanf\_s("%lf", &eps);

sum = summ2(eps);

printf("Сумма элементов последовательности с точностью eps = %lf равна %lf \n", eps, sum);

break;

case '3':

printf("Введите количество членов последовательности n = ");

scanf\_s("%d", &n);

printf("Исключать каждый элемент k = ");

scanf\_s("%d", &k);

print(n, k);

break;

case '4':

printf("Введите eps = ");

scanf\_s("%lf", &eps);

n = findFirstElement(eps);

printf("Номер первого элемента последовательности, для которого выполняется |a(i)| <= eps : i = %d \n", n);

break;

case '5':

printf("Введите eps = ");

scanf\_s("%lf", &eps);

n = findFirstNegativeElement(eps);

printf("Номер первого отрицательного элемента последовательности, для которого выполняется |a(i)| <= eps : i = %d \n", n);

break;

case '6':

break;

}

}