Министерство цифрового развития, связи и массовых 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 №25

Выполнил:

студент группы БСУ1801 Убушуев Б.С

Проверил:

старший преподаватель кафедры ИБ

Барков В. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2021

**Цель**

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

Вариант №25

Рисунок 1 – Исходная формула

**Задание 1**

Написать функцию для вычисления выражения .

Функция в качестве параметра принимает значение 𝑛 и вычисляет сумму 𝑛 слагаемых.

Прототип функции double summ(int n).

Функция не должна использовать функции консольного ввода-вывода.

Листинг 1 – Исходный код файла task1.1.c

#include <math.h>

double summ(int n)

{

double result = 0;

for (int i = 0; i < n-1; ++i)

{

result = result + pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

}

return result;

}

Листинг 2 – Исходный код файла task1.2.c

#include <math.h>

double summ(int n)

{

double result = 0;

{int i = 0;

while (i < n-1)

{

result += pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

++i;

}

}

return result;

}

Листинг 3 – Исходный код файла task1.3.c

#include <math.h>

double summ(int n)

{

double result = 0;

{int i = 0;

if (i < n-1 )

{

do

{

result += pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

++i;

} while (i < n-1 );

}

}

return result;

}



Рисунок 2 – Пример работы функции summ

**Задание 2**

Написать функцию для вычисления выражения с точностью 𝜀.

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

Прототип функции double summ2(double eps).

Функция не должна использовать функции консольного ввода-вывода.

Листинг 4 – Исходный код файла task2.1.c

#include <math.h>

double summ2(double eps)

{

double result = 0;

for (int i = 0; fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)))>eps; ++i)

{

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

}

return result;

}

Листинг 5 – Исходный код файла task2.2.c

#include <math.h>

double summ2(double eps)

{

double result = 0;

{int i = 0;

while (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)))>eps)

{

result += pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

++i;

}

}

return result;

}

Листинг 6 – Исходный код файла task2.3.c

#include <math.h>

double summ2(double eps)

{

double result = 0;

{int i = 0;

if (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)))> eps)

{

do

{

result += pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

i++;

} while (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps);

i++;

}

}

return result;

}



Рисунок 2 – Пример работы функции summ2

**Задание 3**

Написать функцию, которая печатает 𝑛 членов последовательности {}, исключая из неё каждый 𝑘-ый член.

Числа 𝑛 и 𝑘 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

Листинг 7 – Исходный код файла task3.1.c

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double result= 0;

int c = 0;

for (int i = 0; i < (n - 1); ++i)

{

result = pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

if (i == c \* k)

{

c++;

continue;

}

else

{

printf(" Result: %f\n", result);

}

}

}

Листинг 8 – Исходный код файла task3.2.c

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double result= 0;

int c = 0;

{int i = 0;

while (i < (n - 1))

{

result = pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

if (i == c \* k)

{

c++;

++i;

continue;

}

else

{

printf(" Результат: %f\n", result);

}

++i;

}

}

}

Листинг 9 – Исходный код файла task3.3.c

#include <stdio.h>

#include <math.h>

void print(int n, int k)

{

double result= 0;

int c = 0;

{int i = 0;

if (i < n-1)

{

do

{

result = pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i)));

if (i == c\* k)

{

c++;

++i;

continue;

}

else

{

printf(" Result: %f\n", result);

}

++i;

} while (i< n-1);

}

}

}

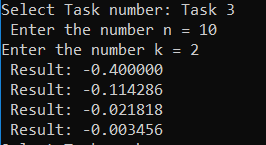


Рисунок 4 – Пример работы функции print

**Задание 4**

Написать функцию, которая возвращает номер первого члена последовательности {𝑎𝑖}, для которого выполняется условие ||≤𝜀.

Функция в качестве параметра принимает значение 𝜀.

Выход из цикла организовать с помощью инструкции перехода break.

Прототип функции int findFirstElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

Листинг 10 – Исходный код файла task4.1.c

#include <math.h>

int findFirstElement(double eps)

{

int result\_i = 0;

for (int i = 0; fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps; i++)

{

if (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps)

{

result\_i = i;

}

else

{

break;

}

}

return ++result\_i;

}

Листинг 11 – Исходный код файла task4.2.c

#include <math.h>

int findFirstElement(double eps)

{

int i = 0;

while (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps)

{

if (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps)

{

i++;

}

else

{

break;

}

}

return i;

}

Листинг 12 – Исходный код файла task4.3.c

#include <math.h>

int findFirstElement(double eps)

{

int i = 0;

if (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps)

{

do

{

if (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps)

{

++i;

}

else

{

break;

}

} while (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps);

}

return i;

}

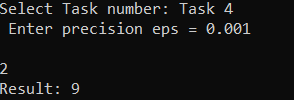


Рисунок 5 – Пример работы функции findFirstElement

**Задание 5**

Написать функцию, которая возвращает номер первого отрицательного члена последовательности {}, для которого выполняется условие ||≤𝜀.

Функция в качестве параметра принимает значение 𝜀.

Выход из цикла организовать с помощью инструкции перехода return.

Прототип функции int findFirstNegativeElement(double eps).

Функция не должна использовать функции консольного ввода-вывода.

Листинг 13 – Исходный код файла task5.1.c

#include <math.h>

int findFirstNegativeElement(double eps)

{

int result\_i = 0;

for (int i = 0; (fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps) && ((pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) < 0)); ++i)

{

result\_i = i;

}

return ++result\_i;

}

Листинг 14 – Исходный код файла task5.2.c

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i = 0;

while ((fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps) && ((pow pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) < 0)) - 1)

{

i++;

}

return ++i;

}

Листинг 15– Исходный код файла task5.3.c

#include <math.h>

int findFirstNegativeElement(double eps)

{

int i = 0;

if ((fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps) && ((pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) < 0)) - 1)

{

do

{

i++;

} while ((fabs(pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) > eps) && ((pow(-1,i) \* ((i + 1) / (pow(3, i) + pow(2, i))) < 0)) - 1);

}

return i;

}

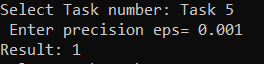


Рисунок 6 – Пример работы функции findFirstNegativeElement

Листинг 16 – Исходный код файла func.h

#ifndef FUNC\_H

#define FUNC\_H

extern double eps;

extern int n, k;

//Task 1

double summ(int n);

//Task 2

double summ2(double eps);

//Task 3

void print(int n, int k);

//Task 4

int findFirstElement(double esp);

//Task 5

int findFirstNegativeElement(double esp);

#endif

Листинг 17 – Исходный код файла task6.1.c

[#include](https://vk.com/im?sel=308961973&st=%23include) <stdio.h>  
[#include](https://vk.com/im?sel=308961973&st=%23include) <math.h>  
[#include](https://vk.com/im?sel=308961973&st=%23include) <conio.h>  
[#include](https://vk.com/im?sel=308961973&st=%23include) <stdlib.h>  
[#include](https://vk.com/im?sel=308961973&st=%23include) "func.h"

int main(void)

{

int z = 0;

setlocale(LC\_CTYPE, "rus");

printf("Menu:\n");

printf("1. Task 1\n");

printf("2. Task 2\n");

printf("3. Task 3\n");

printf("4. Task 4\n");

printf("5. Task 5\n");

printf("6. Exit \n");

do

{

printf("Select Task number: ");

switch (\_getch())

{

case '1':

{

printf("\nTask 1: ");

int n;

float f;

printf("Enter the number n = ");

scanf("%i", &n);

f = summ(n);

printf("Result : %f\n", f);

z = 1;

break;

}

case '2':

{

printf("Task 2:\n ");

float f1;

float eps;

printf("Enter precision eps = ");

scanf("%f", &eps);

f1 = summ2(eps);

printf("Result: %4f\n", f1);

z = 2;

break;

}

case '3':

{

printf("Task 3\n ");

int n1;

int k;

printf("Enter the number n = ");

scanf("%i", &n1);

printf("Enter the number k = ");

scanf("%i", &k);

print(n1, k);

z = 3;

break;

}

case '4':

{

printf("Task 4\n ");

int i;

float eps;

printf("Enter precision eps = ");

scanf("%f\n", &eps);

i = findFirstElement(eps);

printf("Result: %i\n", i);

z = 4;

break;

}

case '5':

{

printf("Task 5\n ");

int i;

float eps;

printf("Enter precision eps= ");

scanf("%f\n", &eps);

i = findFirstNegativeElement(eps);

printf("Result: %i\n", i);

z = 5;

break;

}

case '6':

{

printf("The program has ended.\n ");

z = 6;

break;

}

}

} while (z != 6);

system("pause");

return 0;

}

Листинг 18 – Исходный код файла task6.2.c

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <stdlib.h>

#include "func.h"

int main(void)

{

int z = 0;

printf("Menu:\n");

printf("1. Task 1\n");

printf("2. Task 2\n");

printf("3. Task 3\n");

printf("4. Task 4\n");

printf("5. Task 5\n");

printf("6. Exit \n");

do

{

printf("Select Task number: ");

switch (\_getch())

{

case '1':

{

printf("\nTask 1: ");

int n;

float f;

printf("Enter the number n = ");

scanf("%i", &n);

f = summ(n);

printf("Result : %f\n", f);

z = 1;

break;

}

case '2':

{

printf("Task 2:\n ");

float f1;

float eps;

printf("Enter precision eps = ");

scanf("%f", &eps);

f1 = summ2(eps);

printf("Result: %4f\n", f1);

z = 2;

break;

}

case '3':

{

printf("Task 3\n ");

int n1;

int k;

printf("Enter the number n = ");

scanf("%i", &n1);

printf("Enter the number k = ");

scanf("%i", &k);

print(n1, k);

z = 3;

break;

}

case '4':

{

printf("Task 4\n ");

int i;

float eps;

printf("Enter precision eps = ");

scanf("%f\n", &eps);

i = findFirstElement(eps);

printf("Result: %i\n", i);

z = 4;

break;

}

case '5':

{

printf("Task 5\n ");

int i;

float eps;

printf("Enter precision eps= ");

scanf("%f\n", &eps);

i = findFirstNegativeElement(eps);

printf("Result: %i\n", i);

z = 5;

break;

}

case '6':

{

printf("The program has ended.\n ");

z = 6;

break;

}

}

} while (z != 6);

system("pause");

return 0;

}

Листинг 19 – Исходный код файла task6.3.c

#include <stdio.h>

#include <math.h>

#include <conio.h>

#include <stdlib.h>

#include "func.h"

int main(void)

{

int z = 0;

printf("Menu:\n");

printf("1. Task 1\n");

printf("2. Task 2\n");

printf("3. Task 3\n");

printf("4. Task 4\n");

printf("5. Task 5\n");

printf("6. Exit \n");

do

{

printf("Select Task number: ");

switch (\_getch())

{

case '1':

{

printf("\nTask 1: ");

int n;

float f;

printf("Enter the number n = ");

scanf("%i", &n);

f = summ(n);

printf("Result : %f\n", f);

z = 1;

break;

}

case '2':

{

printf("Task 2:\n ");

float f1;

float eps;

printf("Enter precision eps = ");

scanf("%f", &eps);

f1 = summ2(eps);

printf("Result: %4f\n", f1);

z = 2;

break;

}

case '3':

{

printf("Task 3\n ");

int n1;

int k;

printf("Enter the number n = ");

scanf("%i", &n1);

printf("Enter the number k = ");

scanf("%i", &k);

print(n1, k);

z = 3;

break;

}

case '4':

{

printf("Task 4\n ");

int i;

float eps;

printf("Enter precision eps = ");

scanf("%f\n", &eps);

i = findFirstElement(eps);

printf("Result: %i\n", i);

z = 4;

break;

}

case '5':

{

printf("Task 5\n ");

int i;

float eps;

printf("Enter precision eps= ");

scanf("%f\n", &eps);

i = findFirstNegativeElement(eps);

printf("Result: %i\n", i);

z = 5;

break;

}

case '6':

{

printf("The program has ended.\n ");

z = 6;

break;

}

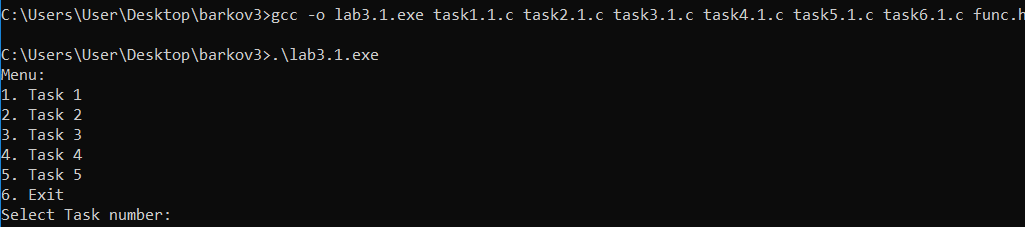
}

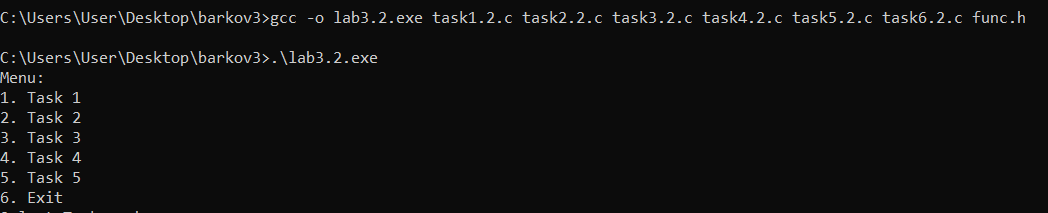
} while (z != 6);

system("pause");

return 0;

}

****

****

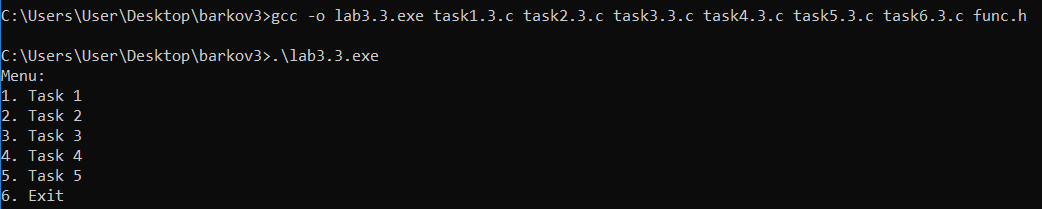
****

Рисунок 17 – Пример работы функции main

https://github.com/BasanUbshuevBsu1801/barkovlab3

**Вывод**

В данной работе были разобраны аспекты работы с функциями for, while и do while. Так же были разобраны операторы continue и break.