

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное
образовательное учреждение высшего образования

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
СВЯЗИ И ИНФОРМАТИКИ**

Кафедра информационной безопасности

ЛАБОРАТОРНАЯ РАБОТА

№ 3

по дисциплине «ООПСЗИ»

«Регулярные и итерационные циклы»

Выполнил

студент группы БП31801

Игумнов В.О.

Проверил

Барков В. В.

Москва, 2021

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 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. Прототипы функции вынести в заголовочный файл. В решении создать три проекта.

Индивидуальный вариант задания:

$$a_i = (-1)^i \frac{2^{i+1}}{2^{2i} + 1}$$

Выполнение:

Задание 1.

Написать функцию для вычисления выражения $\sum_{i=0}^{n-1} a_i$. Функция в качестве параметра принимает значение **n** и вычисляет сумму *n* слагаемых.

Прототип функции **double sum(int n)**.

Функция не должна использовать функции консольного ввода-вывода.

1. Создал отдельный файл для кода функции индивидуального задания с названием **individual_func.c**.

Листинг 1 – Исходный код файла `individual_func.c`

```
GNU nano 4.8      individual_func.c
#include <math.h>

double a(int i) {
    return pow(-1, i) * (pow(2, i+1) / (pow(2, 2*i) + 1));
}
```

2. Написал функции для вычисления выражения в трёх вариантах с использованием **for**, **while**, **do ... while**.

Листинг 2.1 – Исходный код файла task1_for.c

```

GNU nano 4.8 task1_for.c
#include "task1.h"

double sum(int n) {
    static double result = 0;
    for (int i = 0; i < n; ++i) {
        result += a(i);
    }
    return result;
}

```

Листинг 2.2 – Исходный код файла task1_while.c

```

GNU nano 4.8 task1_while.c
#include "task1.h"

double sum(int n) {
    static double result = 0;
    int i = 0;
    while (i < n) {
        result += a(i);
        ++i;
    }
    return result;
}

```

Листинг 2.3 – Исходный код файла task1_do_while.c

```

GNU nano 4.8 task1_do_while.c
#include "task1.h"

double sum(int n) {
    static double result = 0;
    int i = 0;
    do {
        result += a(i);
        ++i;
    } while (i < n);
    return result;
}

```

Задание 2.

Написать функцию для вычисления выражения $\sum_{i=0}^{n-1} a_i$ с точность ε . Функция в качестве параметра принимает значение ε и вычисляет сумму до тех пор, пока не выполнится условие $|a_i| \leq \varepsilon$.

Прототип функции **double sum2(double eps).**

Функция не должна использовать функции консольного ввода-вывода.

Листинг 3.1 – Исходный код файла task2_for.c

```
GNU nano 4.8 task2_for.c
#include <math.h>
#include "../task1.h"

double sum2(double eps) {
    static double result = 0;
    for (int i = 0; ; ++i) {
        if (fabs(a(i)) <= eps) {
            break;
        }
        else {
            result += a(i);
        }
    }
    return result;
}
```

Листинг 3.2 – Исходный код файла task2_while.c

```
GNU nano 4.8 task2/task2_while.c Изменён
#include <math.h>
#include "../task1.h"

double sum2(double eps) {
    static double result = 0;
    static int i = 0;
    while (fabs(a(i)) > eps) {
        result += a(i);
        ++i;
    }
    return result;
}
```

Листинг 3.1 – Исходный код файла task2_do_while.c

```
GNU nano 4.8 task2/task2_do_while.c
#include <math.h>
#include "../task1.h"

double sum2(double eps) {
    static double result = 0;
    static int i = 0;
    do {
        result += a(i);
        ++i;
    } while (fabs(a(i)) > eps);
    return result;
}
```

Задание 3.

Написать функцию, которая печатает n членов последовательности $\{a_n\}$, 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).**

Листинг 4.1 – Исходный код файла task3_for.c

```
GNU nano 4.8 task3/task3_for.c
#include <stdio.h>
#include "../task1.h"

void print(int n, int k) {
    for (int i = 1; i <= n; ++i) {\
        if (!(i % k)) {
            continue;
        }
        printf("%lf ", a(i));
    }
}
```

Листинг 4.2 – Исходный код файла task3_while.c

```
GNU nano 4.8 task3/task3_while.c
#include <stdio.h>
#include "../task1.h"

void print(int n, int k) {
    static int i = 1;
    while (i <= n) {
        if (!(i % k)) {
            i++;
            continue;
        }
        printf("%lf ", a(i));
        i++;
    }
}
```

Листинг 4.3 – Исходный код файла task3_while_do.c

```
GNU nano 4.8 task3/task3_do_while.c
#include <stdio.h>
#include "../task1.h"

void print(int n, int k) {
    static int i = 1;
    do {
        if (!(i % k)) {
            i++;
            continue;
        }
        printf("%lf ", a(i));
        i++;
    } while (i < n);
}
```

Задание 4.

Написать функцию, которая возвращает номер первого члена последовательности $\{a_i\}$, для которого выполняется условие $|a_i| \leq \varepsilon$.

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 5.1 – Исходный код файла task4_for.c

```
GNU nano 4.8 task4_for.c
#include <math.h>
#include "../task1.h"
#include <stdio.h>

int findFirstElement(double eps) {
    static double result = 0;
    static int i = 0;
    for (;;) {
        if (fabs(a(i)) <= eps) {
            break;
        }
        else {
            result += a(i);
        }
    }
    return i;
}
```

Листинг 5.2 – Исходный код файла task4_while.c

```
GNU nano 4.8 task4_while.c
#include <math.h>
#include "../task1.h"

int findFirstElement(double eps) {
    static double result = 0;
    static int i = 0;
    while (fabs(a(i)) > eps) {
        result += a(i);
        ++i;
    }
    return i;
}
```

```

GNU nano 4.8      task4_do_while.c
#include <math.h>
#include "../task1.h"

int findFirstElement(double eps) {
    static double result = 0;
    static int i = 0;
    do {
        result += a(i);
        ++i;
    } while (fabs(a(i)) > eps);
    return i;
}

```

Задание 5.

Написать функцию, которая возвращает номер первого отрицательного члена последовательности $\{a_i\}$, для которого выполняется условие $|a_i| \leq \varepsilon$.

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.

```

GNU nano 4.8      task5/task5_for.c
#include <math.h>
#include "../task1.h"

int findFirstNegativeElement(double eps) {
    static int i = 0;
    for (;;) {
        if (a(i) < 0 && fabs(a(i)) <= eps) {
            return i;
        }
        ++i;
    }
}

```

```

GNU nano 4.8      task5/task5_while.c
#include <math.h>
#include "../task1.h"

int findFirstNegativeElement(double eps) {
    static int i = 0;
    while ("true") {
        ++i;
        if (a(i) < 0 && fabs(a(i)) <= eps) { return i; }
    }
}

```

```

GNU nano 4.8      task5/task5_do_while.c
#include <math.h>
#include "../task1.h"

int findFirstNegativeElement(double eps) {
    static int i = 0;
    do {
        if (a(i) < 0 && fabs(a(i)) <= eps) { return i; }
        ++i;
    } while ("true");
}

```

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 выполнять 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.


```

1  #include <stdio.h>
2  #include "task1.h"
3  #include <string.h>
4  #include <stdlib.h>
5  #include <conio.h>
6
7  int main() {
8      static int n, k;
9      static double eps;
10     while ("true") {
11         system("cls");
12         printf("1.Exercise 1\n2.Exercise 2\n3.Exercise 3\n4.Exercise 4\n5.Exercise 5\n6. Quit\n");
13         switch(getchar()) {
14             case '1':
15                 system("cls");
16                 printf("Exercise 1.\nType int 'n'\n");
17                 scanf("%d", &n);
18                 printf("Ans: %lf\n", sum(n));
19                 printf("type for exit ");
20                 while (! _kbhit() ) { continue; }
21                 break;
22             case '2':
23                 system("cls");
24                 printf("Exercise 2.\nType 0 < e < 1 'eps'\n");
25                 scanf("%lf", &eps);
26                 printf("Ans: %lf\n", sum2(eps));
27                 printf("type for exit ");
28                 while (! _kbhit() ) { continue; }
29                 break;
30             case '3':
31                 system("cls");
32                 printf("Exercise 3.\nType int 'n' and 'k'\n");
33                 scanf("%d %d", &n, &k);
34                 printf("Ans: \n");
35                 print(n, k);
36                 printf("type for exit ");
37                 while (! _kbhit() ) { continue; }
38                 break;
39             continue;
40             case '4':
41                 system("cls");
42                 printf("Exercise 4.\nType 0 < e < 1 'eps'\n");
43                 scanf("%lf", &eps);
44                 printf("Ans: %d\n", findFirstElement(eps));
45                 printf("type for exit ");
46                 while (! _kbhit() ) { continue; }
47                 break;
48             continue;
49             case '5':
50                 system("cls");
51                 printf("Exercise 5.\nType 0 < e < 1 'eps'\n");
52                 scanf("%lf", &eps);
53                 printf("Ans: %d\n", findFirstNegativeElement(eps));
54                 printf("type for exit ");
55                 while (! _kbhit() ) { continue; }
56                 break;
57             case '6':
58                 return 0;
59             }
60     }
61 }

```

Ссылка на Github репозиторий: https://github.com/Cno6/Igumnov_OOP_labs