Министерство Цифрового Развития, Связи и Массовых Коммуникаций Российской Федерации  
Федеральное Государственное Бюджетное Образовательное Учреждение Высшего Образования Ордена Трудового Красного знамени   
«Московский технический университет связи и информатики»

Кафедра информационной безопасности

Лабораторная работа №3

## Разветвляющиеся алгоритмы.

Выполнил  
Студент группы БПЗ1801

Шишкин Иван Ильич

Проверил:

Барков Вячеслав Валерьевич

Москва 2021

**ОБЩЕЕ ЗАДАНИЕ.**

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

По номеру Вашего варианта выбрать задачи 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. Прототип функции вынести в заголовочный файл. В решении создать три проекта.

**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.

**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 |𝑎𝑖|≤𝜀.

Прототип функции double summ2(double eps).

Функция не должна использовать функции консольного ввода-вывода.

**Задание 3.**

Написать функцию, которая печатает 𝑛 членов последовательности {𝑎𝑖}, 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

**Задание 4.**

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

**Задание 5.**

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

**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

**ИНДИВИДУАЛЬНЫЙ ВАРИАНТ ЗАДАНИЯ.**

Вариант 26(6).

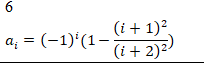


Рисунок 1 – Индивидуальное задание.

**ПРОГРАММНЫЙ КОД.**

**Задание 1.**

For:

#include <iostream>

#include <math.h>

using namespace std;

double summ(int count)

{

double summ = 0;

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

{

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

}

return summ;

}

void main()

{

int count;

cout << "Enter count = ";

cin >> count;

cout << "Function summ = " << summ(count);

}

Листинг 1 – Исходный код файла Задание 1-1.cpp

While:

#include <iostream>

#include <math.h>

using namespace std;

double summ(int count)

{

double summ = 0;

int i = 0;

while (i<=count)

{

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

i++;

}

return summ;

}

Листинг 2 – Исходный код файла Задание 1-2.cpp

Функция void main() совпадает с Листингом 1.

Do While:

#include <iostream>

#include <math.h>

using namespace std;

double summ(int count)

{

double summ = 0;

int i = 0;

do

{

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

i++;

} while (i <= count);

return summ;

}

Листинг 3 – Исходный код файла Задание 1-3.cpp

Функция void main() совпадает с Листингом 1.

**Результаты работы программы:**

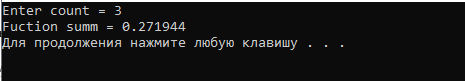


Рисунок 2 – Результат работы программы к заданию 1.

**Задание 2.**

For:

#include <iostream>

#include <math.h>

using namespace std;

double summ(double eps)

{

int i = 0;

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

for (i = 1; abs(summ) > eps; i++)

{

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

}

return summ;

}

void main()

{

double eps;

cout << "Enter eps = ";

cin >> eps;

cout << "Function summ = " << summ(eps);

}

Листинг 4 – Исходный код файла Задание 2-1.cpp

While:

#include <iostream>

#include <math.h>

using namespace std;

double summ(double eps)

{

int i = 0;

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

while (abs(summ) > eps)

{

i++;

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

}

return summ;

}

Листинг 5 – Исходный код файла Задание 2-2.cpp

Функция void main() совпадает с Листингом 4.

Do While:

#include <iostream>

#include <math.h>

using namespace std;

double summ(double eps)

{

int i = 0;

double summ = 0;

do

{

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

i++;

} while (abs(summ) > eps);

return summ;

}

Листинг 6 – Исходный код файла Задание 2-3.cpp

Функция void main() совпадает с Листингом 4.

**Результат работы программы:**

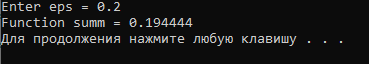


Рисунок 2 – Результат работы программы к заданию 2.

**Задание 3.**

For:

#include <iostream>

#include <math.h>

using namespace std;

void print(int n, int k)

{

double count = 0;

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

{

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

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

{

continue;

}

else

{

cout << count << " ";

}

}

}

void main()

{

int n,k = 0;

cout << "Enter n = ";

cin >> n;

cout << "Enter k = ";

cin >> k;

print(n, k);

}

Листинг 7 – Исходный код файла Задание 3-1.cpp

While:

#include <iostream>

#include <math.h>

using namespace std;

void print(int n, int k)

{

double count = 0;

int i = 0;

while (i < n)

{

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

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

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

}

}

Листинг 8 – Исходный код файла Задание 3-2.cpp

Функция void main() совпадает с Листингом 7.

#include <iostream>

#include <math.h>

using namespace std;

void print(int n, int k)

{

double count = 0;

int i = 0;

do

{

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

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

{

i++;

continue;

}

else

{

cout << count << " ";

}

i++;

} while (i < n);

}

Листинг 9 - Исходный код файла Задание 3-3.cpp

Функция void main() совпадает с Листингом 7.

**Результат работы программы:**

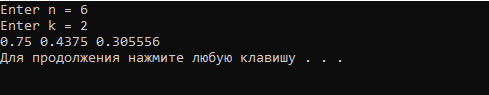


Рисунок 4 – Результат работы программы к заданию 3.

**Задание 4.**

For:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

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

if (abs(count) < eps)

{

break;

}

}

return i+1;

}

void main()

{

double eps;

cout << "Enter eps = ";

cin >> eps;

cout << "Number = " << findFirstElement(eps);

}

Листинг 10 - Исходный код файла Задание 4-1.cpp

While:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

while(true)

{

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

i++;

if (abs(count) < eps)

{

break;

}

}

return i;

}

Листинг 11 - Исходный код файла Задание 4-2.cpp

Функция void main() совпадает с Листингом 10.

Do while:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstElement(double eps)

{

int i = 0;

double count = 0;

do

{

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

i++;

if (abs(count) < eps)

{

break;

}

} while (true);

return i;

}

Листинг 12 - Исходный код файла Задание 4-3.cpp

Функция void main() совпадает с Листингом 10.

**Результат работы программы:**



Рисунок 5 – Результат работы программы к заданию 4.

**Задание 5.**

For:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstNegativeElement(double eps)

{

int i;

double count = 0;

for (i = 0; ; i++)

{

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

if (abs(count) < eps && count < 0)

{

return i + 1;

}

}

}

void main()

{

double eps;

cout << "Enter eps = ";

cin >> eps;

cout << "Number = " << findFirstNegativeElement(eps);

}

Листинг 13 - Исходный код файла Задание 5-1.cpp

While:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

while (true)

{

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

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

}

}

Листинг 14 - Исходный код файла Задание 5-2.cpp

Функция void main() совпадает с Листингом 13.

Do while:

#include <iostream>

#include <math.h>

using namespace std;

int findFirstNegativeElement(double eps)

{

int i = 0;

double count = 0;

do

{

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

i++;

if (abs(count) < eps && count < 0)

{

return i;

}

} while (true);

}

Листинг 15 - Исходный код файла Задание 5-3.cpp

Функция void main() совпадает с Листингом 13.

**Результат работы программы:**



Рисунок 6 – Результат работы программы к заданию 5.

**Задание 6.**

#include <iostream>

#include <math.h>

using namespace std;

void task1();

void task2();

void task3();

void task4();

void task5();

int main()

{

int number;

setlocale(LC\_CTYPE, "rus");

while (true)

{

system("cls");

cout << "Задания ЛР3 с циклом For" << endl

<< "[1] Задание 1 " << endl

<< "[2] Задание 2" << endl

<< "[3] Задание 3" << endl

<< "[4] Задание 4" << endl

<< "[5] Задание 5" << endl

<< "[6] Выход" << endl

<< "Введите номер подпункта: ";

cin >> number;

switch (number)

{

case 1:

{

system("cls");

task1();

cout << endl;

system("pause");

break;

}

case 2:

{

system("cls");

task2();

cout << endl;

system("pause");

break;

}

case 3:

{

system("cls");

task3();

cout << endl;

system("pause");

break;

}

case 4:

{

system("cls");

task4();

cout << endl;

system("pause");

break;

}

case 5:

{

system("cls");

task5();

cout << endl;

system("pause");

break;

}

case 6:

{

return true;

}

}

}

}

Листинг 16 - Исходный код файла Задание 6-1.cpp

Задания 6-1, 6-2, 6-3 имеют 2 файла Задание 6-1.cpp, Задание 6-1 tasks.cpp и так далее. Файлы Задание 6-1….6-3 tasks содержат файлы функций описанных в ранее заданиях, функции main заменены на task1()… task 2()

**Результат работы программы:**

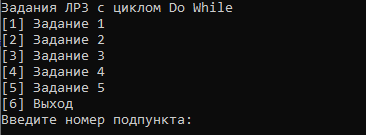


Рисунок 7 – Результат работы программы к заданию 6.