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»

Кафедра «Информационная безопасность»

Лабораторная работа №2

«Модули. Линейные, разветвляющиеся и итерационные алгоритмы»

Вариант 26

учебной дисциплины «Разработка безопасного программного обеспечения»

Выполнила ст. гр. БАС2101:

Ярцева С.С.

Проверил:

Барков В.В.

Москва 2023

# Цель работы

# Овладеть навыками создания модулей в языке C++ и научится создавать линейные, разветвляющиеся и итерационные программы на языке C++.

# Задание

По номеру Вашего варианта выбрать функции , , .

В каждом задании необходимо реализовать функции:

, , , и

При реализации функций и вычисление очередного слагаемого организовать с помощью функции .

Вычисление значения фукнции следует производить до тех пор, пока не будет достигнута заданная точность , т.е. до тех пор, пока истинно условие .

Разработанные фукнции разместить в отдельном модуле. Имя модуля должно состоять из идентификатора student, номера студенческого билета исполнителя, номера лабораторной работы, номера варианта и номера задания. Все части названия модуля разделить точкой. Например, для задания 1 варианта 9: student.1bib21000.Lab2.Variant9.Task1.

В модуле определить пространство имен с названием дисциплины (RBPO). Внутри пространства имен RBPO определить пространство имен, включающее номер лабораторной работы (Lab2). Внутри пространства имен RBPO::Lab2 определить пространство имен с номером варианта (VariantM, где M – номер варианта). Внутри пространства имён RBPO::Lab2::VariantM определить пространство имен с номером задания (TaskN, где N – номер задания).

Все разработанные функции в задании N должны лежать в пространстве имен RBPO::Lab2::VariantM::TaskN, где M – номер варианта, N – номер задания.

***Задание 1***

При реализации функции использовать условное выражение (не инструкцию выбора if).

Функции и реализовать с помощью цикла for.

**Определения** функций , , , , поместить в одну единицу трансляции, описывающую **интерфейс** **модуля** (Module Interface Unit – файл с расширением \*.ixx).

Экспортировать функции , , , . Само пространство имен НЕ экспортировать.

При необходимости подключить заголовочные файлы, используйте фрагмент глобального модуля (Global Module Fragment).

***Задание 2***

При реализации функции использовать инструкцию выбора if.

Функции и реализовать с помощью цикла while.

**Объявления** функций , , , поместить в единицу трансляции, описывающую **интерфейс** **модуля** (Module Interface Unit). Объявление функции помещать не нужно.

Экспортировать пространство имен RBPO::Lab2::Task2 целиком.

**Определения** функций , f , , поместить в **одну** единицу трансляции, описывающую **реализацию** **модуля** (Module Implementation Unit).

При необходимости подключить заголовочные файлы, используйте используйте фрагмент глобального модуля (Global Module Fragment).

***Задание 3***

Функции и реализовать с помощью цикла do … while. Реализации функций должны возвращать те же результаты, что и в предыдущих заданиях.

**Объявления** функций , , , , поместить в единицу трансляции, описывающую **интерфейс** **модуля** (Module Interface Unit).

Экспортировать функции , , , . Само пространство имен НЕ экспортировать.

**Каждое определение** функций , f , , поместить в **отдельную** единицу трансляции, описывающую **реализацию** **модуля** (Module Implementation Unit).

Итоговый модуль будет содержать одну единицу трансляции, описывающую интерфейс модуля и 5 единиц трансляции, описывающих реализацию модуля. В каждой единице трансляции будет размещено одно определение функции.

При необходимости подключить заголовочные файлы, используйте фрагмент глобального модуля (Global Module Fragment).

***Задание 4***

Модуль разделить на 5 разделов (module partition).

**Объявление каждой из** функций , , , , поместить в **отдельную** единицу трансляции, описывающую **интерфейс раздела** **модуля** (Module Partition Interface Unit).

Экспортировать функции , , , в соответствующих единицах трасляции.

В единице трансляции, описывающей **интерфейс модуля** (Module Interface Unit) импортировать разделы, содержащие экспорт фукнций , , , и экспортировать их.

**Каждое определение** функций , f , , поместить в **отдельную** единицу трансляции, описывающую **реализацию раздела** **модуля** (Module Partition Implementation Unit). В единицах трансляции, содержащих реализации и потребуется импортировать раздел с объявлением функции .

Итоговый модуль будет содержать одну единицу трансляции, описывающую интерфейс модуля и 5 разделов. Каждый из разделов будет включать единицу трансляции, описывающую интерфейс раздела модуля (в которой будет содержаться прототип функции), и единицу трансляции, описывающую реализацию раздела модуля (в которой будет содержаться реализация фукнции).

При необходимости подключить заголовочные файлы, используйте фрагмент глобального модуля (Global Module Fragment).

***Задание 5***

**Объявления и определения** функций , , , , поместить в одну единицу трансляции, описывающую **интерфейс** **модуля** (Module Interface Unit – файл с расширением \*.ixx).

При этом **определение** (реализацию) функций поместить в **приватный фрагмент модуля** (Module Private Fragment).

Экспортировать функции , , , . Само пространство имен НЕ экспортировать.

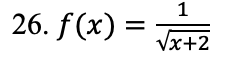
При необходимости подключить заголовочные файлы, используйте фрагмент глобального модуля (Global Module Fragment).

***Задание 6***

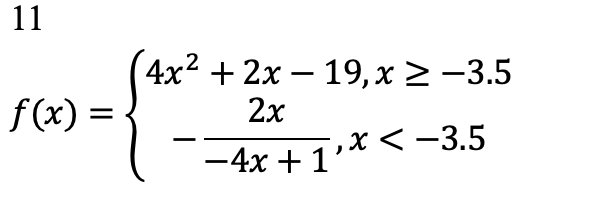
Разработать функцию main, демонстрирующую работу всех разработанных функций в заданиях 1-5.

Функция должна в цикле показывать меню и давать возможность выбрать задание и продемонстрировать работу всех функций из этого задания.

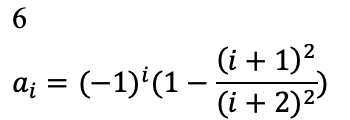
Вариант 26.  
Функция f1:



Функция f2:



Функция a:



**Код**

**Файл student\_1bib21065\_Lab2\_Variant26\_Task1.ixx**

module;  
#include <cmath>  
export module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
export import :A;  
export import :F1;  
export import :F2;  
export import :F3;  
export import :F4;  
  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task1 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 export namespace Task2 {  
 double f1(double);  
 double f2(double);  
 double f3(unsigned long);  
 double f4(double);  
 };  
 namespace Task3 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 namespace Task4 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 namespace Task5 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 namespace Task1 {  
 double f1(double x)  
 {  
 double result = 1 / (sqrt(x + 2));  
 return result;  
 }  
 double f2(double x)  
 {  
 double result;  
 if (x >= -3.5) {  
 result = 4 \* pow(x, 2) + 2 \* x - 19;  
 }  
 else  
 result = -2 \* x / (-4 \* x + 1);  
  
 return result;  
 }  
 double a(long n)  
 {  
 double result = (-1) ^ n \* (1 - ((n + 1) ^ 2 / (n + 2) ^ 2));  
 return result;  
 }  
 double f3(unsigned long n)  
 {  
 double result = 0;  
 for (unsigned long i = 0; i < n; i++) {  
 result += a(i);  
 }  
 return result;  
 }  
 double f4(double epsilon)  
 {  
 double result = 0;  
 double n = 8.98847e+307;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 if (abs(a(i) - a(i - 1)) > epsilon) {  
 return result;  
 }  
 }  
 return 0;  
 };  
 };  
 };  
 };  
};

**Файл student\_1bib21065\_Lab2\_Variant26\_Task2.ixx**

module;  
#include <cmath>  
  
module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task2 {  
 double f1(double);  
 double f2(double);  
 double f3(unsigned long);  
 double f4(double);  
 double a(long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task2::f1(double x) {  
 return 1 / (sqrt(x + 2));  
};  
  
double RBPO::Lab2::Variant26::Task2::f2(double x) {  
 double result;  
 if (x >= -3.5) {  
 result = 4 \* pow(x, 2) + 2 \* x - 19;  
 }  
 else  
 result = -2 \* x / (-4 \* x + 1);  
  
 return result;  
};  
  
double RBPO::Lab2::Variant26::Task2::a(long n) {  
 return (-1) ^ n \* (1 - ((n + 1) ^ 2 / (n + 2) ^ 2));  
};  
  
double RBPO::Lab2::Variant26::Task2::f3(unsigned long n) {  
 double result = 0;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 }  
 return result;  
};  
  
double RBPO::Lab2::Variant26::Task2::f4(double epsilon) {  
 double result = 0;  
 double n = 8.98847e+307;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 if (abs(a(i) - a(i - 1)) > epsilon) {  
 return result;  
 }  
 }  
 return 0;  
};

**Файл student\_1bib21065\_Lab2\_Variant26\_Task3.ixx**

module;  
#include <cmath>  
  
module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3{  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 };  
 };  
};

**Файл student\_1bib21065\_Lab2\_Variant26\_Task4.ixx**

module;  
#include <cmath>  
export module student\_1bib21065\_Lab2\_Variant26\_Task1;  
export import :A;  
export import :F1;  
export import :F2;  
export import :F3;  
export import :F4;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task5 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 };  
 };  
};

**Файл student\_1bib21065\_Lab2\_Variant26\_Task5.ixx**

module;  
#include <cmath>  
export module student\_1bib21065\_Lab2\_Variant26\_Task1;  
export import :A;  
export import :F1;  
export import :F2;  
export import :F3;  
export import :F4;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task5 {  
 export double f1(double);  
 export double f2(double);  
 export double f3(unsigned long);  
 export double f4(double);  
 double a(long);  
 };  
 };  
 };  
};  
  
module :private;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task5 {  
 double f1(double x) {  
 double pi = 3.14;  
 return 1 / (sqrt(x + 2));  
 };  
  
 double f2(double x) {  
 double result;  
 if (x >= -3.5) {  
 result = 4 \* pow(x, 2) + 2 \* x - 19;  
 }  
 else  
 result = -2 \* x / (-4 \* x + 1);  
  
 return result;  
 };  
  
 double f3(unsigned long n) {  
 double result = 0;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 }  
 return result;  
 };  
  
 double f4(double epsilon) {  
 double result = 0;  
 double n = 8.98847e+307;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 if (abs(a(i) - a(i - 1)) > epsilon) {  
 return result;  
 }  
 }  
 return 0;  
 }  
  
 double a(long n) {  
 return (-1) ^ n \* (1 - ((n + 1) ^ 2 / (n + 2) ^ 2));  
 };  
 };  
 };  
 };  
};

**Файл ModuleA.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1:A;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double a(long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task4::a(long n) {  
 return (-1) ^ n \* (1 - ((n + 1) ^ 2 / (n + 2) ^ 2));  
};

**Файл ModuleF1.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1:F1;  
#include <cmath>  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double f1(double);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task4::f1(double x) {  
 return 1 / (sqrt(x + 2));  
};

**Файл ModuleF2.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1:F2;  
#include <cmath>  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double f2(double);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task4::f2(double x) {  
 double result;  
 if (x >= -3.5) {  
 result = 4 \* pow(x, 2) + 2 \* x - 19;  
 }  
 else  
 result = -2 \* x / (-4 \* x + 1);  
  
 return result;  
};

**Файл ModuleF3.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1:F3;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double f3(unsigned long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task4::f3(unsigned long n) {  
 double result = a(0);  
 for (unsigned long i = 0; i < n; result += a(i++)) {  
 result += a(i);  
 }  
 return result;  
};

**Файл ModuleF4.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1:F4;  
#include <cmath>  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double f4(double);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task4::f4(double epsilon) {  
   
 double result = 0;  
 double n = 8.98847e+307;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 if (abs(a(i) - a(i - 1)) > epsilon) {  
 return result;  
 }  
 }  
 return 0;  
};

**Файл a.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3 {  
 double a(long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task3::a(long n) {  
 return (-1) ^ n \* (1 - ((n + 1) ^ 2 / (n + 2) ^ 2));  
};

**Файл f1.cpp**

module;  
#include <cmath>  
  
module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3 {  
 double f1(double);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task3::f1(double x) {  
 return 1 / (sqrt(x + 2));  
}

**Файл f2.cpp**

module;  
#include <cmath>  
  
module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3 {  
 double f2(double);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task3::f2(double x) {  
 double result;  
 if (x >= -3.5) {  
 result = 4 \* pow(x, 2) + 2 \* x - 19;  
 }  
 else  
 result = -2 \* x / (-4 \* x + 1);  
  
 return result;  
};

**Файл f3.cpp**

module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3 {  
 double f3(unsigned long);  
 double a(long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task3::f3(unsigned long n) {  
 double result = 0;  
 for (unsigned long i = 0; i < n; i++) {  
 result += a(i);  
 }  
 return result;  
};

**Файл f4.cpp**

module;  
#include <cmath>  
module student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task3 {  
 double f4(double);  
 double a(long);  
 };  
 };  
 };  
};  
  
double RBPO::Lab2::Variant26::Task3::f4(double epsilon) {  
 double result = 0;  
 double n = 8.98847e+307;  
 for (int i = 0; i < n; i++) {  
 result += a(i);  
 if (abs(a(i) - a(i - 1)) > epsilon) {  
 return result;  
 }  
 }  
 return 0;  
};

**Файл A.ixx**

export module student\_1bib21065\_Lab2\_Variant26\_Task1:A;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 double a(long);  
 };  
 };  
 };  
};

**Файл F1.ixx**

export module student\_1bib21065\_Lab2\_Variant26\_Task1:F1;  
import "cmath";  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 export double f1(double);  
 };  
 };  
 };  
};

**Файл F2.ixx**

export module student\_1bib21065\_Lab2\_Variant26\_Task1:F2;  
import "cmath";  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 export double f2(double);  
 };  
 };  
 };  
};

**Файл F3.ixx**

export module student\_1bib21065\_Lab2\_Variant26\_Task1:F3;  
  
import :A;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 export double f3(unsigned long);  
 };  
 };  
 };  
};

**Файл F4.ixx**

export module student\_1bib21065\_Lab2\_Variant26\_Task1:F4;  
import "cmath";  
import :A;  
  
namespace RBPO {  
 namespace Lab2 {  
 namespace Variant26 {  
 namespace Task4 {  
 export double f4(double);  
 };  
 };  
 };  
};

**Файл main.cpp**

#include <iostream>  
using namespace std;  
  
import student\_1bib21065\_Lab2\_Variant26\_Task1;  
  
  
int main() {  
 char c;  
 do  
 {  
 cout <<"For first task input :'1'"  
 <<"\nFor second task input :'2'"  
 <<"\nFor third task input:'3'"  
 <<"\nFor fourth task input:'4'"  
 <<"\nFor fifth task input :'5'"  
 <<"\nFor end the session input:'0'" <<endl;  
 cin >> c;  
 cout <<endl;  
 switch (c) {  
 unsigned long n; double x, eps;  
 case '1':  
 cout << "Input x, n and eps:" << endl;  
 cin >> x >> n >> eps;  
 cout <<endl<< "f1(x) = " << RBPO::Lab2::Variant26::Task1::f1(x) <<endl  
 << "f2(x) = " << RBPO::Lab2::Variant26::Task1::f2(x) <<endl  
 << "f3(n) = " << RBPO::Lab2::Variant26::Task1::f3(n) <<endl  
 << "f4(eps) = " << RBPO::Lab2::Variant26::Task1::f4(eps) <<endl << endl;  
 break;  
 case '2':  
 cout << "Input x, n and eps:" << endl;  
 cin >> x >> n >> eps;  
 cout << endl << "f1(x) = " << RBPO::Lab2::Variant26::Task2::f1(x) << endl  
 << "f2(x) = " << RBPO::Lab2::Variant26::Task2::f2(x) << endl  
 << "f3(n) = " << RBPO::Lab2::Variant26::Task2::f3(n) << endl  
 << "f4(eps) = " << RBPO::Lab2::Variant26::Task2::f4(eps) <<endl << endl;  
 break;  
 case '3':  
 cout <<"Input x, n and eps:" << endl;  
 cin >> x >> n >> eps;  
 cout << endl << "f1(x) = " << RBPO::Lab2::Variant26::Task3::f1(x) << endl  
 << "f2(x) = " << RBPO::Lab2::Variant26::Task3::f2(x) << endl  
 << "f3(n) = " << RBPO::Lab2::Variant26::Task3::f3(n) << endl  
 << "f4(eps) = " << RBPO::Lab2::Variant26::Task3::f4(eps) <<endl << endl;  
 break;  
 case '4':  
 cout <<"Input x, n and eps:" << endl;  
 cin >> x >> n >> eps;  
 cout << endl << "f1(x) = " << RBPO::Lab2::Variant26::Task4::f1(x) << endl  
 << "f2(x) = " << RBPO::Lab2::Variant26::Task4::f2(x) << endl  
 << "f3(n) = " << RBPO::Lab2::Variant26::Task4::f3(n) << endl  
 << "f4(eps) = " << RBPO::Lab2::Variant26::Task4::f4(eps) << endl << endl;  
 break;  
 case '5':  
 cout << "Input x, n and eps:" << endl;  
 cin >> x >> n >> eps;  
 cout << endl << "f1(x) = " << RBPO::Lab2::Variant26::Task5::f1(x) << endl  
 << "f2(x) = " << RBPO::Lab2::Variant26::Task5::f2(x) << endl  
 << "f3(n) = " << RBPO::Lab2::Variant26::Task5::f3(n) << endl  
 << "f4(eps) = " << RBPO::Lab2::Variant26::Task5::f4(eps) <<endl<<endl;  
 break;  
 case '0':  
 break;  
 default:  
 cout << "Inputed data is incorrect!" << endl;  
 break;  
 }  
 } while (c != '0');  
 return 0;  
};

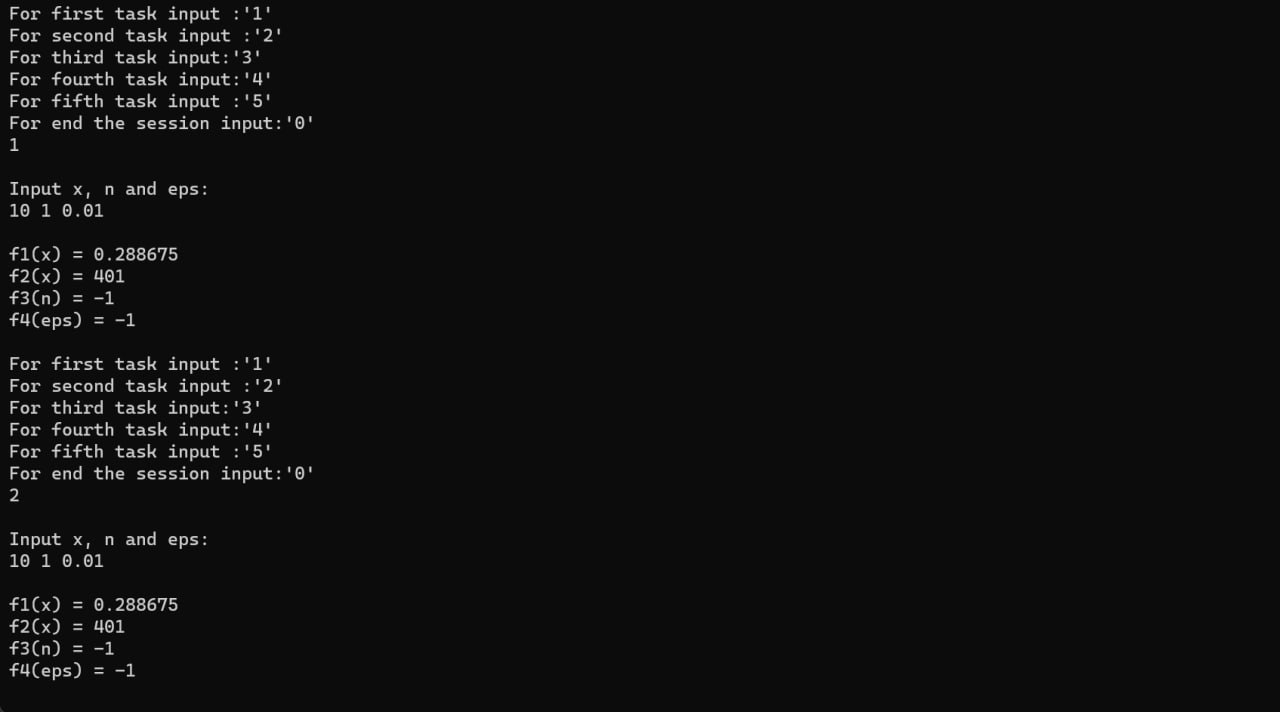
****

Рис. 1. Вывод программы

**Вывод**

# Приобретены навыки создания модулей в языке C++ и создания линейных, разветвляющиеся и итерационные программ на языке C++.