Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное образовательное учреждение высшего образования

**«Пермский национальный исследовательский политехнический университет»**

Электротехнический факультет

Кафедра «Информационные технологии и автоматизированные системы» направление подготовки: 09.03.04 – «Программная инженерия»

**Лабораторная работа №3.**

**«Рекурсивная функция»**

Выполнил студент гр. РИС-24-3б

Гладков Ярослав Алексеевич

Проверил:

Доцент. каф. ИТАС

Ольга Андреевна Полякова

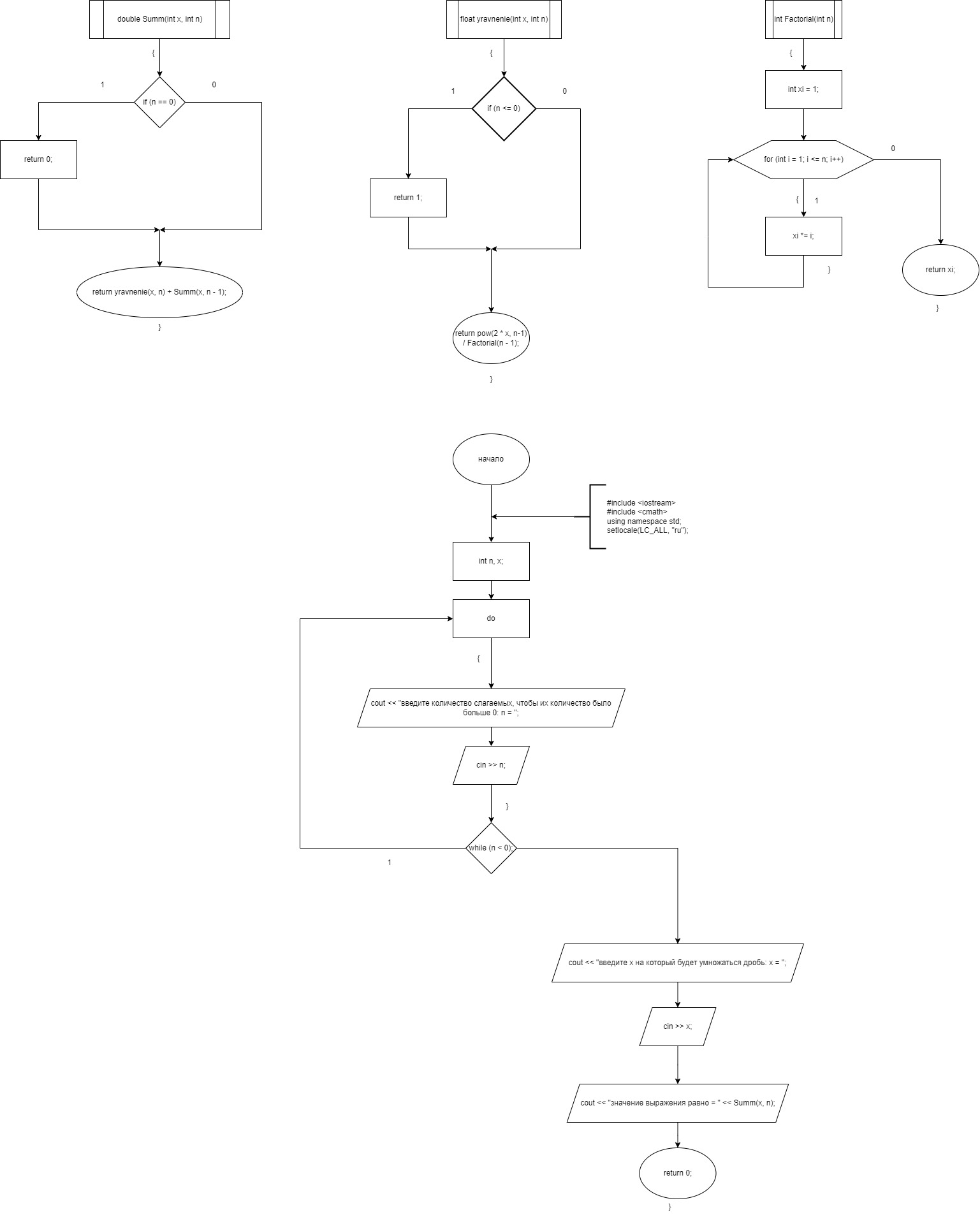
(оценка) (подпись)

(дата)

г. Пермь, 2024

Написать программу с рекурсивной функцией решающую уравнение

**Блок схема:**



**Код программы:**

#include <iostream>

#include <cmath>

using namespace std;

int Factorial(int n)

{

int xi = 1;

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

{

xi \*= i;

}

return xi;

}

float yravnenie(int x, int n)

{

if (n <= 0) {

return 1;

}

return pow(2 \* x, n-1) / Factorial(n - 1);

}

double Summ(int x, int n)

{

if (n == 0)

{

return 0;

}

else

{

return yravnenie(x, n) + Summ(x, n - 1);

}

}

int main()

{

setlocale(LC\_ALL, "ru");

int n, x;

do

{

cout << "введите количество слагаемых, чтобы их количество было больше 0: n = ";

cin >> n;

} while (n < 0);

cout << "введите x на который будет умножаться дробь: x = ";

cin >> x;

cout << "значение выражения равно = " << Summ(x, n);

return 0;

}

**Как работает код:**

1. Summ(6, 5):  
   - Вызывается yravnenie(6, 5):  
     - pow(2 \* 6, 4) / Factorial(4)  
     - pow(12, 4) = 20736  
     - Factorial(4) = 24  
     - Результат: 20736 / 24 = 864  
   - Затем вызывается Summ(6, 4).  
  
2. Summ(6, 4):  
   - Вызывается yravnenie(6, 4):  
     - pow(2 \* 6, 3) / Factorial(3)  
     - pow(12, 3) = 1728  
     - Factorial(3) = 6  
     - Результат: 1728 / 6 = 288  
   - Затем вызывается Summ(6, 3).  
  
3. Summ(6, 3):  
   - Вызывается yravnenie(6, 3):  
     - pow(2 \* 6, 2) / Factorial(2)  
     - pow(12, 2) = 144  
     - Factorial(2) = 2  
     - Результат: 144 / 2 = 72  
   - Затем вызывается Summ(6, 2).  
  
4. Summ(6, 2):  
   - Вызывается yravnenie(6, 2):  
     - pow(2 \* 6, 1) / Factorial(1)  
     - pow(12, 1) = 12  
     - Factorial(1) = 1  
     - Результат: 12 / 1 = 12  
   - Затем вызывается Summ(6, 1).  
  
5. Summ(6, 1):  
   - Вызывается yravnenie(6, 1):  
     - pow(2 \* 6, 0) / Factorial(0)  
     - pow(12, 0) = 1  
     - Factorial(0) = 1  
     - Результат: 1 / 1 = 1  
   - Затем вызывается Summ(6, 0).  
  
6. Summ(6, 0):  
   - Возвращает 0.

Сумма результатов:

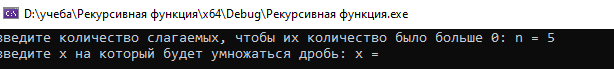
1. Summ(6, 1) = 1
2. Summ(6, 2) = 12 + 1 = 13
3. Summ(6, 3) = 72 + 13 = 85
4. Summ(6, 4) = 288 + 85 = 373
5. Summ(6, 5) = 864 + 373 = 1237

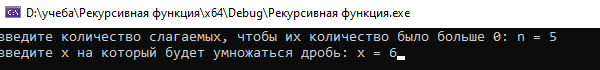
Итоговое значение выражения = 1237

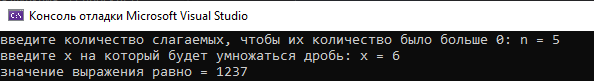
**Работа программы:**











Git hub ссылка:

<https://github.com/YaroslavRi/Recursia.git>