Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЁТ**

**по лабораторной работе №3**

Дисциплина: основы алгоритмизации и программирования

Тема: методы решения нелинейных уравнений

Вариант 23

Выполнил:

Студент группы ИВТ-20-2б

Брейкин Алексей Дмитриевич

Проверил:

Доцент кафедры ИТАС

Полякова О. А.

**Пермь, 2020**

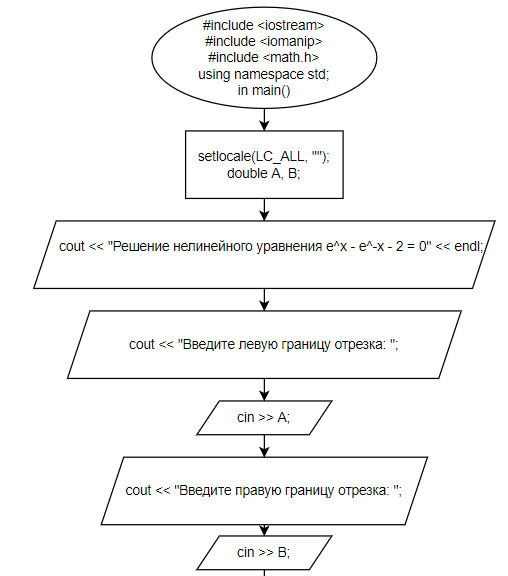
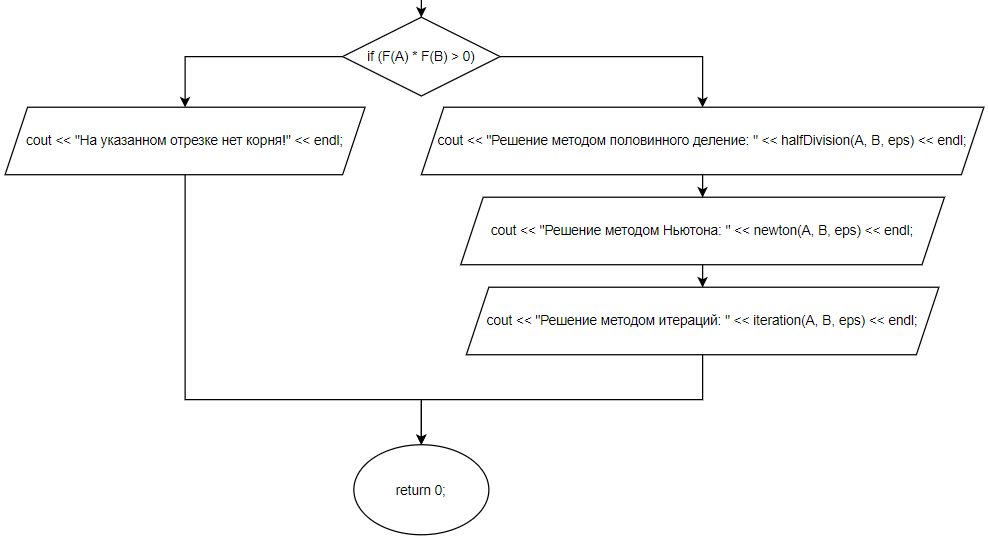
**Постановка задачи**

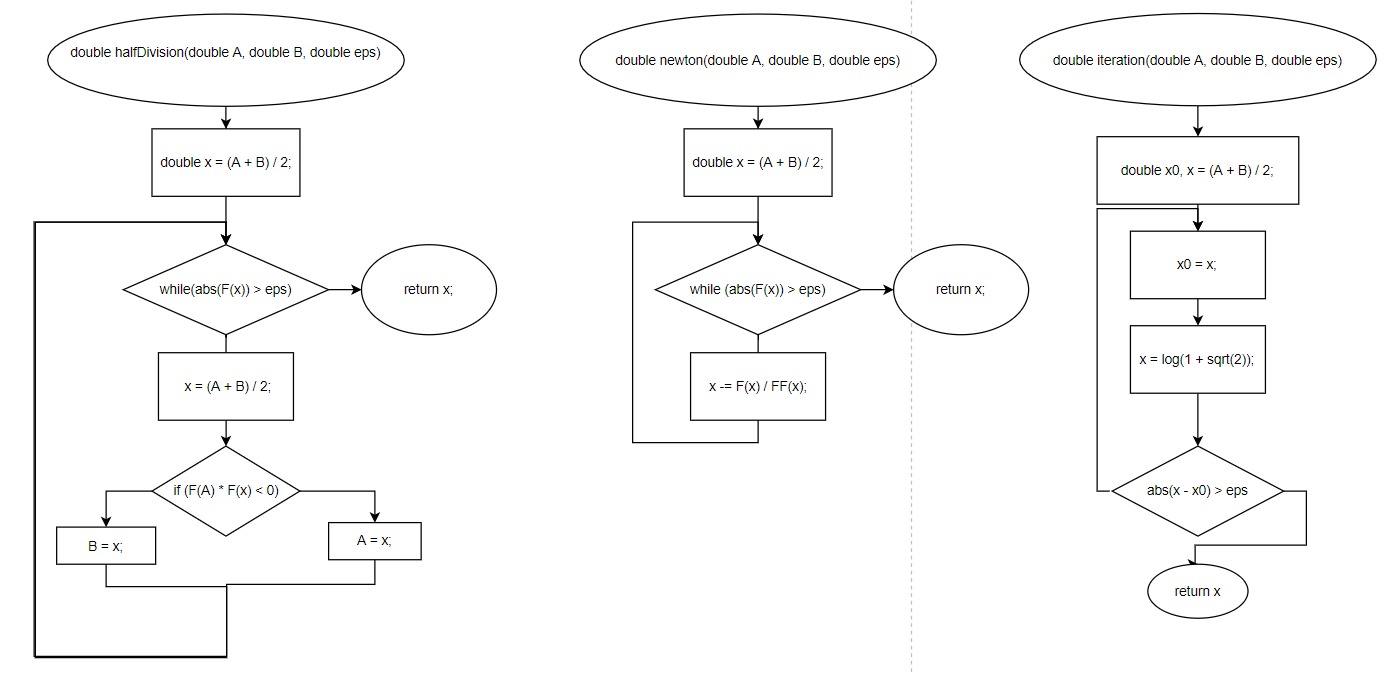
Реализовать методы половинного деления, Ньютона, итераций для решения нелинейного уравнения:

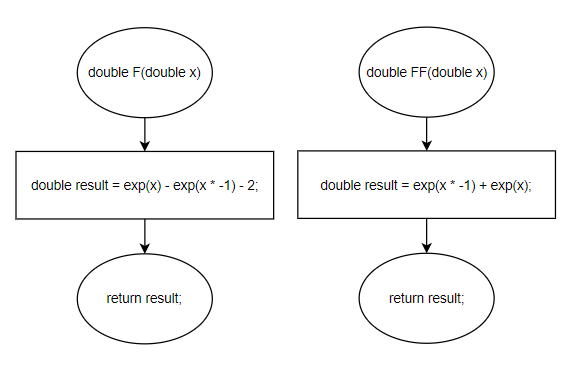
**Анализ задачи**

1. Попросим пользователя ввести левое и правое ограничения отрезка, на котором будет производится корень уравнения.
2. Реализуем метод половинного деления, для этого возьмём произвольную точку и будем проверять знак слева и справа от значения функции в этой точке: если знак отрицательный, то исключаем левый интервал, и правый, если положительный. Дальше берём середину выбранного интервала и снова повторяем данные действия, пока интервал не станет достаточно малым
3. Реализуем метод Ньютона, для этого возьмём точку Х, которая будет серединой отрезка между левым и правым ограничением и с помощью цикла, пока значение функции от Х не станет достаточно малым, будем вычитать из Х отношение значения функции к значению производной этой функции в данной точке;
4. Выведем на экран все значения, полученные с помощью реализованных методов, описанных выше.

**Блок-схема**





**Исходный код**

#include <iostream>

#include <iomanip>

#include <math.h>

#include <windows.h>

using namespace std;

inline double F(double x) { return exp(x) - exp(x \* -1) - 2; }

inline double FF(double x) { return exp(x \* -1) + exp(x); } // производная

double halfDivision(double A, double B, double eps)

{

double x = (A + B) / 2;

while (abs(F(x)) > eps) {

x = (A + B) / 2;

if (F(A) \* F(x) < 0)

B = x;

else

A = x;

}

return x;

}

double newton(double A, double B, double eps)

{

double x = (A + B) / 2;

while (abs(F(x)) > eps) {

x -= F(x) / FF(x);

}

return x;

}

double iteration(double A, double B, double eps)

{

double x0, x = (A + B) / 2;

do

{

x0 = x;

x = log(1 + sqrt(2));

} while (abs(x - x0) > eps);

return x;

}

int main()

{

SetConsoleOutputCP(CP\_UTF8);

cout << "Решение нелинейного уравнения e^x - e^-x - 2 = 0" << endl;

double A, B;

cout << "Введите левую границу отрезка: ";

cin >> A;

cout << "Введите правую границу отрезка: ";

cin >> B;

double eps = 0.000001; // 10 ^ -6

if (F(A) \* F(B) > 0) {

cout << "На указанном отрезке нет корня!" << endl;

return 0;

}

  cout << fixed << setprecision(4);

cout << "Решение методом половинного деление: " << halfDivision(A, B, eps) << endl;

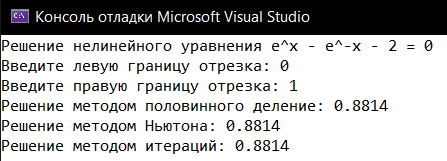
cout << "Решение методом Ньютона: " << newton(A, B, eps) << endl;

cout << "Решение методом итераций: " << iteration(A, B, eps) << endl;

return 0;

}

**Скриншоты работы программы**

****