Министерство науки и высшего образования Российской Федерации Федеральное государственное автономное учреждение высшего образования «Пермский национальный исследовательский политехнический университет»

ПНИПУ

**Лабораторная работа  
“Методы решения нелинейных уравнений”**

Выполнил:   
студент группы РИС-23-1б   
Карпов Иван Васильевич

Проверила:   
доцент кафедры ИТАС   
О.А. Полякова

Пермь 2023 г.

**Условие:** решить тремя методами уравнение Отрезок, содержащий корень: [0;1] Точное значение: 0,8814

**Анализ задачи:**

**Метод половинного деления:**

Задаем начальные значения левой и правой границ отрезка: a = 0, b = 1.

Проверяем условие f(a) \* f(b) < 0, где f(x) =.

Если условие не выполняется, выходим из метода, так как на данном отрезке нет корней.

Иначе, выполняем итерации метода:

Вычисляем среднюю точку отрезка: x0 = (a + b) / 2.

Проверяем условие f(x0) = 0 или |b - a| <= epsilon.

Если условие выполняется, выводим значение x0 как приближенный корень.

Иначе, проверяем знак f(a) \* f(c):

Если f(a) \* f(x0) < 0, обновляем правую границу b = x0.

Иначе, обновляем левую границу a = x0.

Возвращаемся к шагу 4 и продолжаем итерации до достижения заданной точности или нахождения корня.

**Метод Ньютона:**

Задаем начальное приближение x0 = 0.5 (можно выбрать любое значение из отрезка [0, 1]).

Выполняем итерации метода:

На каждой итерации вычисляем значение функции и ее производной в точке x: f(x) и f '(x).

Обновляем значение x по формуле: x = x - f(x) / f '(x).

Проверяем условие |x – x-1| < epsilon. Если условие выполняется, выводим значение x как приближенный корень.

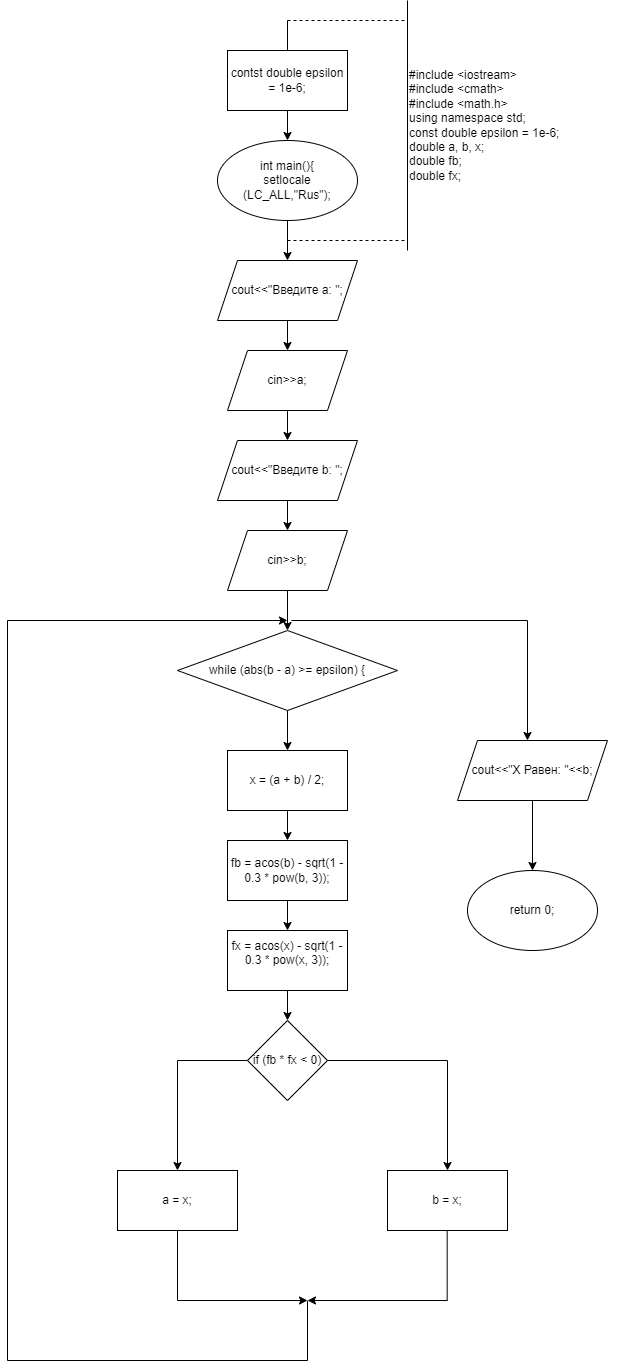
Иначе, возвращаемся к шагу 2 и продолжаем итерации до достижения заданной точности или нахождения корня.

**Метод итераций:**

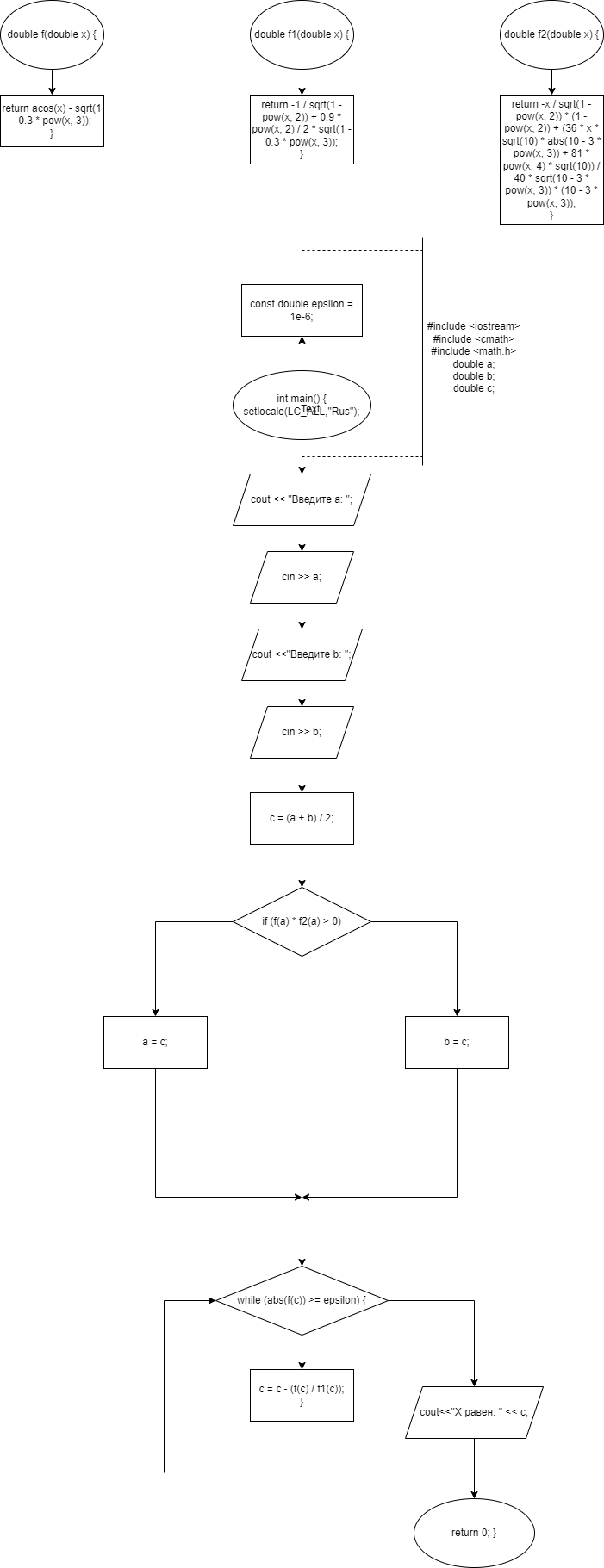
1. Представляем исходное уравнение f(x)=0 в виде x=µ(x)
2. На отрезке [a; b] выбираем любую точку
3. Следующее приближенное значение вычисляется по формуле xi=µ(xi-1)
4. 3 шаг повторять, пока |xi-xi-1| >
5. Если |xi-xi-1| <= цикл заканчивается, xi – приближенный корень

**Блок схемы:**

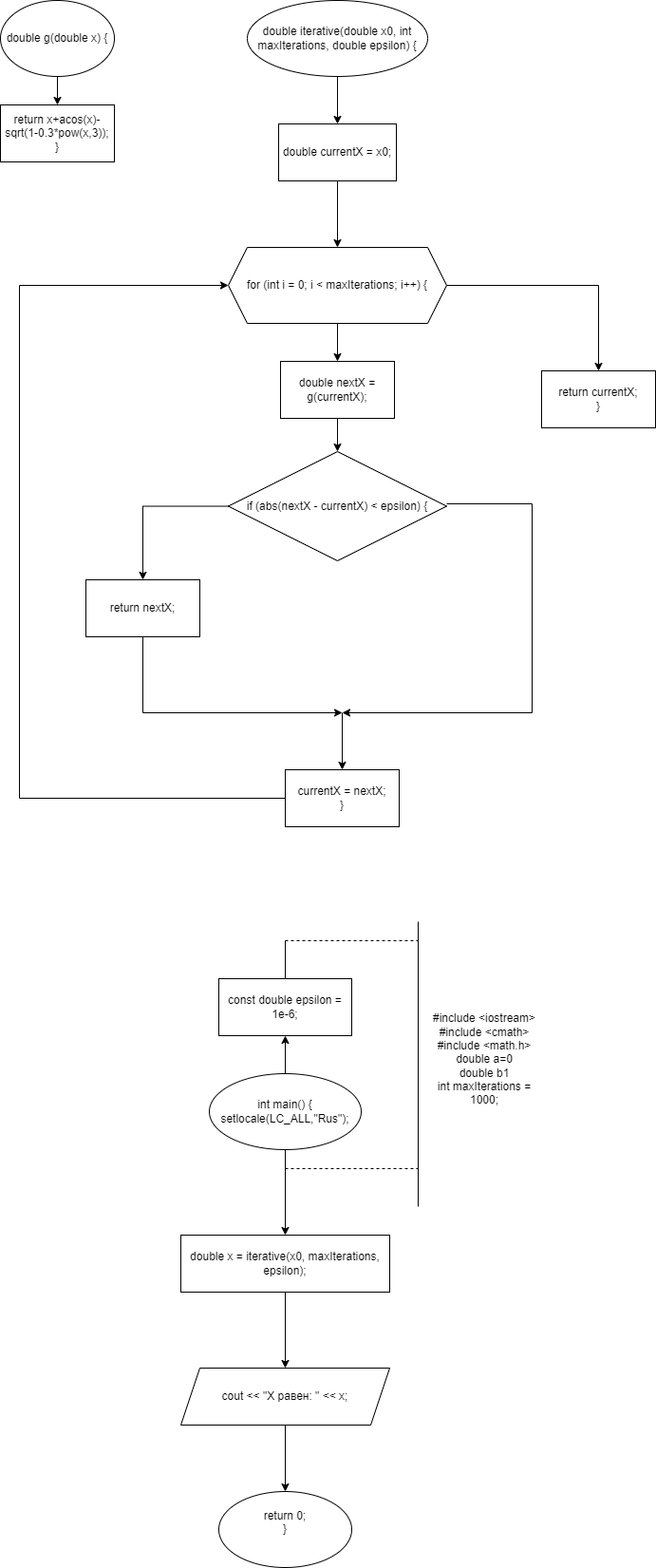
**Метод половинного деления:**

****

**Метод Ньютона:**

****

**Метод итераций:**



**Код на С++:  
1) Метод половинного деления:**

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 1e-6;

int main() {

setlocale(LC\_ALL, "Rus");

double a, b, x, fb, fx;

int coun = 0;

cout << "Введите левую границу: ";

cin >> a;

cout << "Введите правую границу: ";

cin >> b;

while (abs(b - a) >= epsilon) {

x = (a + b) / 2;

cout << x << endl;

coun += 1;

fb = pow(2.7183, b) - pow(2.7183, -b) - 2;

fx = pow(2.7183, x) - pow(2.7183, -x) - 2;

if (fb \* fx < 0)

a = x;

else

b = x;

}

cout << "Шаги: " << coun << endl;

cout << "Приблизительный корень уравнения: " << x;

return 0;

}

**2) Метод Ньютона:**

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 1e-6;

double f(double x) {

return pow(2.7183, x) - pow(2.7183, -x) - 2;

}

double f1(double x) {

return pow(2.7183, x) + pow(2.7183, -x);

}

double f2(double x) {

return pow(2.7183, x) - pow(2.7183, -x);

}

int main() {

setlocale(LC\_ALL, "Rus");

double a, b, x;

int coun = 0;

cout << "Введите левую границу: ";

cin >> a;

cout << "Введите правую границу: ";

cin >> b;

x = (a + b) / 2;

if (f(a) \* f2(a) > 0)

a = x;

else

b = x;

while (abs(f(x)) >= epsilon) {

x = x - (f(x) / f1(x));

coun += 1;

cout << x << endl;

}

cout << "Приблизительный корень уравнения: " << x << " Шаги: " << coun;

return 0;

}

**3) Метод Итераций:**

#include <iostream>

#include <cmath>

using namespace std;

const double epsilon = 1e-6;

double f(double x) {

return x - 0.3932 \* (pow(2.7183, x) - pow(2.7183, -x) - 2);

}

double iterative(double x0, int maxIterations, double epsilon) {

double currentX = x0;

int coun = 0;

for (int i = 0; i < maxIterations; i++) {

double nextX = f(currentX);

cout << nextX << endl;

coun += 1;

if (abs(nextX - currentX) < epsilon) {

cout << "Шаги: " << coun << endl;

return nextX;

}

currentX = nextX;

}

cout << "Шаги: " << coun << endl;

return currentX;

}

int main() {

setlocale(LC\_ALL, "Rus");

double a, b;

int maxIterations = 1000;

cout << "Введите левую границу: ";

cin >> a;

cout << "Введите правую границу: ";

cin >> b;

double x0 = (a + b) / 2;

double x = iterative(x0, maxIterations, epsilon);

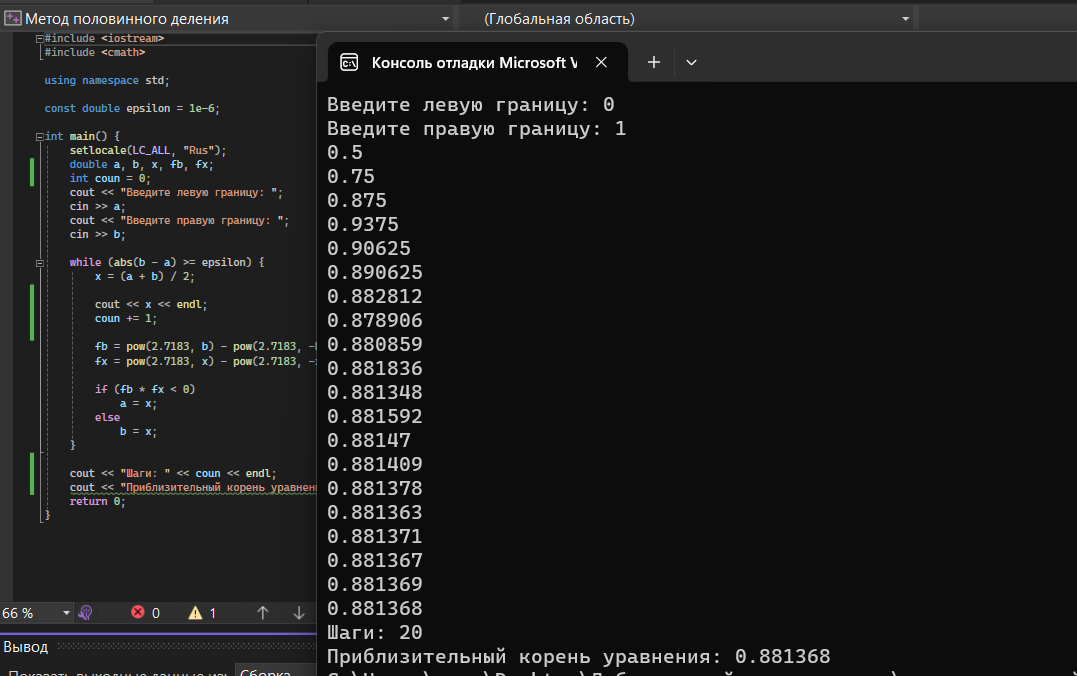
cout << "Приблизительный корень уравнения: " << x;

return 0;

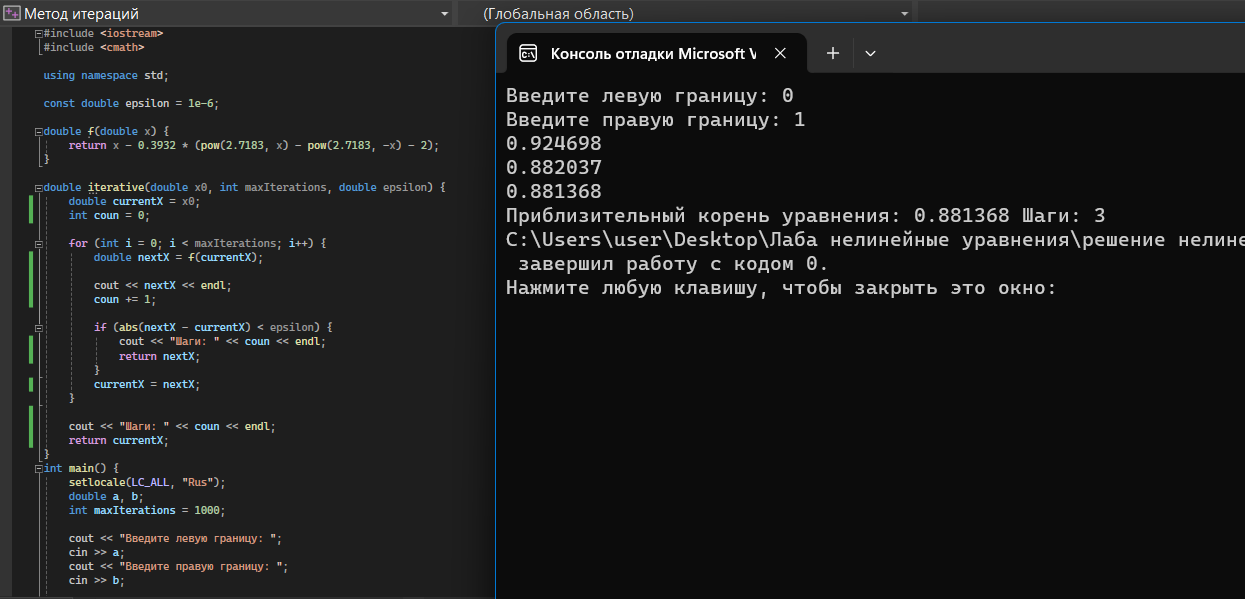
}

**Работа программы:**

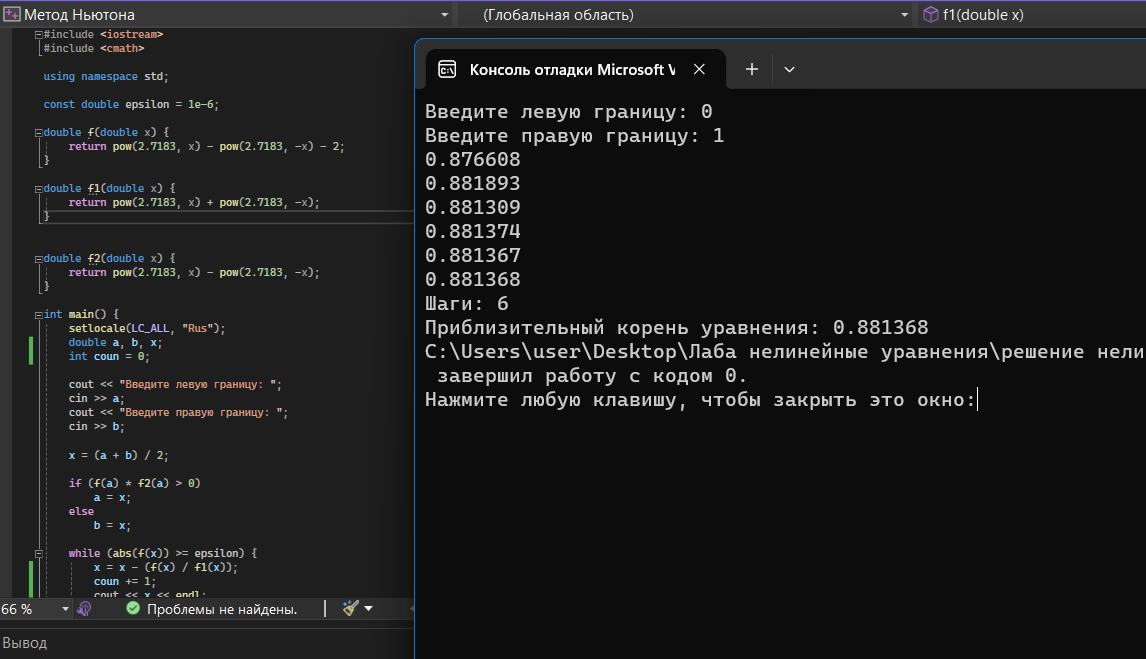
**1) Метод половинного деления:**

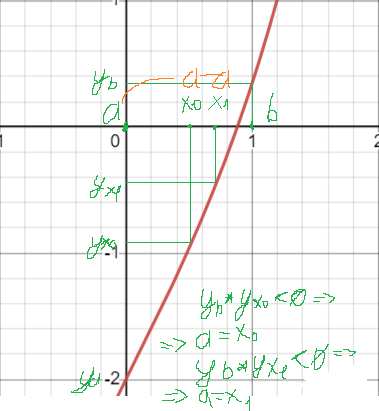


**2) Метод Ньютона:**

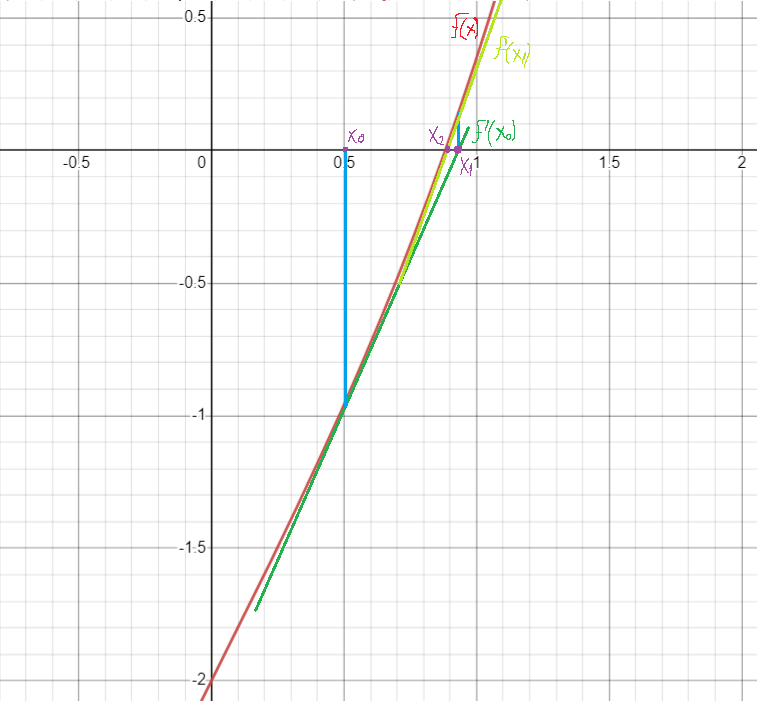
****

**3) Метод Итераций:**

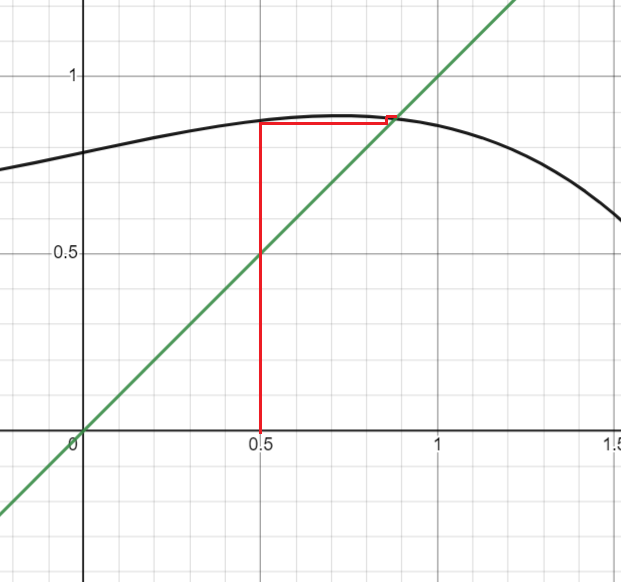


**Графики:**  
1) Метод половинного деления:  


2)Метод Ньютона:



3)Метод Итераций:



**Выводы:** Задачи были выполнили. Всё получилось, программа сработала корректно и вывела желаемый результат.