RegulaFalsi.cpp

```cpp
#include<iostream>
#include<cmath>
#include <iomanip>
#include<fstream>
#include <cstdlib>

using namespace std;

double f(double x)
{
    return pow(x,10.0)-1;
}

double RegulaFalsi(double xl, double xu, double es, double max_itr)
{
    ofstream outfile("RegulaFalsi1.dat"); // Open file to write output
    outfile << "Iteration" << setw(15) << "xl" << setw(15) << "xu" << setw(15) << "xr" << setw(15) << "ea" << endl;

    int itr = 0;
    double ea = 100,xr = 0.0,xrold;
    double fl = f(xl), fu = f(xu);
    int iu = 0;
    int il = 0;

    while(true)
    {
        xrold = xr;
        xr = xu - ((fu*(xl-xu))/(fl-fu));
        itr = itr + 1;
        if(xr !=0)
        {
            ea = fabs((xr - xrold)/(xr))*100;
        }
        double test = fl*f(xr);
        if(test<0)
        {
            xu = xr;
            fu = f(xu);
            iu = 0;
            il++;
            if(il>=2)
            {
                fl = fl/2;
            }
        }
        else if(test>0)
        {
            xl = xr;
            fl = f(xl);
            il = 0;
            iu++;
```

```cpp
                if(iu>=2)
                {
                    fu = fu/2;
                }
            }
            else
            {
                ea = 0;
            }
        cout << itr << setw(18) << xl << setw(18) << xu << setw(18) << xr << setw(18) << ea
    << endl;
        outfile << itr << setw(18) << xl << setw(18) << xu << setw(18) << xr << setw(18) <<
    ea << endl;
        if(ea <= es && max_itr)
        {
            break;
        }
    }
    return xr;
}


int main()
{
    double xl,xu,es,max_itr;
    cout << "Enter the value of lower bound 'xl'\n";
    cin >> xl;
    cout << "Enter the value of lower bound 'xu'\n";
    cin >> xu;
    cout << "Enter the value of lower bound 'es'\n";
    cin >> es;
    cout << "Enter the value of maximum iterations 'max_itr'\n";
    cin >> max_itr;

    while(f(xl)*f(xu)>=0)
    {
        cout << "Invalid bracket! f(xl) * f(xu) must be negative.\n";
        cout << "Re-enter lower bound (xl): ";
        cin >> xl;
        cout << "Re-enter upper bound (xu): ";
        cin >> xu;
    }

    double root = RegulaFalsi(xl,xu,es,max_itr);
    cout << "Approximate root is :" << root << endl;
    system("python vis.py");

    return 0;
}
```