Title: Finding integration of $\frac{1}{1+x}$ by numerical methods Trapezoidal Rule, Simpson's 1/3-Rule & Simpson's 3/8-Rule.

Theoretical Background:

Let the interval $[a, b]$ be divided into n equal subintervals such that $a = x_0 < x_1 < x_2 < \cdots x_n = b$. Clearly, $x_n = x_0 + nh$. Hence the integral becomes,

$$I = \int_{x_1}^{x_0} ydx$$

On simplification,

$$\int_{x_1}^{x_0} ydx = nh[y_0 + \frac{n}{2}\Delta y_0 + \frac{n(2n-3)}{12}\Delta^2 y_0 + n(n-2)^2\Delta^3 y_0 + \cdots]$$

Trapezoidal Rule: Setting $n = 1$ and simplifying,

$$\int_{x_1}^{x_0} ydx = \frac{h}{2}[y_0 + 2(y_1 + y_2 + \cdots + y_{n-1}) + y_n]$$

Simpson's 1/3-Rule: Setting $n = 2$ and simplifying,

$$\int_{x_1}^{x_0} ydx = \frac{h}{3}[y_0 + 4(y_1 + y_3 + y_5 + \cdots + y_{n-1}) + 4(y_2 + y_4 + y_6 + \cdots + y_{n-2}) + y_n]$$

Simpson's 3/8-Rule: Setting $n = 3$ and simplifying,

$$\int_{x_1}^{x_0} ydx = \frac{3h}{8}[(y_0 + 3y_1 + 3y_2 + y_3) + (y_3 + 3y_4 + 3y_5 + y_6) + \cdots + (y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n)]$$

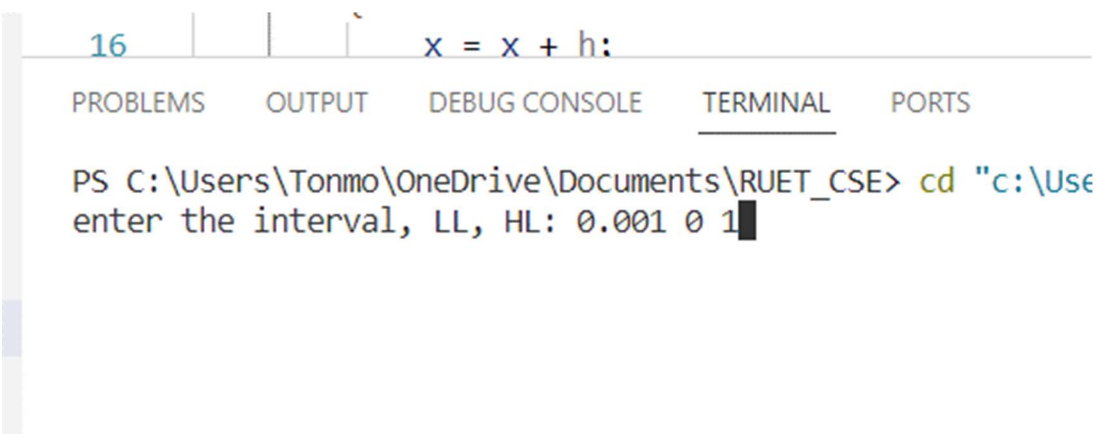
Program:

```
#include <iostream>
using namespace std;
double funk(double x)
{
    return 1 / (1 + x);
}

double integration(double h, int y, int a, int b)
{
    double n = ((b - a) / h), x = a, sum = funk(x);
    for (int i = 1; i <= n; i++)
    {
        if (y == 1)
        {
            x = x + h;
            cout << x << " " << funk(x) << endl;
            sum += 2 * funk(x);
        }
        else if (y == 2)
        {
            if (i % 2 == 0)
            {
                x = x + h;
                cout << x << " " << funk(x) << endl;
                sum += 2 * funk(x);
            }
            else if (i % 2 != 0)
            {
                x = x + h;
                cout << x << " " << funk(x) << endl;
                sum += 4 * funk(x);
            }
        }
        else if (y == 3)
        {
            if (i % 3 == 0)
            {
                x = x + h;
                cout << x << " " << funk(x) << endl;
                sum += 2 * funk(x);
            }
            else if (i % 3 != 0)
            {
```

```
                x = x + h;
                cout << x << " " << funk(x) << endl;
                sum += 3 * funk(x);
            }
        }
    }
    if (y == 1)
        sum = (h / 2) * (sum + funk(x + n));
    else if (y == 2)
        sum = (h / 3) * (sum + funk(x + n));
    else if (y == 3)
        sum = (3 * h / 8) * (sum + funk(x + n));
    return sum;
}
int main()
{
    double h, a, b;
    cout << "enter the interval, LL, HL: ";
    cin >> h >> a >> b;
    double re = integration(h, 1, a, b);
    double re2 = integration(h, 2, a, b);
    double re3 = integration(h, 3, a, b);
    cout << "The area value of 1/(1+x) by trapizoidal is = " << re << endl;
    cout << "The area value of 1/(1+x) by simpson's 1/3 rule is = " << re2 <<
endl;
    cout << "The area value of 1/(1+x) by simpson's 3/8 rule is = " << re3 <<
endl;
}
```

Input & Output:



```
0.997 0.500751
0.998 0.500501
0.999 0.50025
1 0.5
The area value of 1/(1+x) by trapizoidal is = 0.693398
The area value of 1/(1+x) by simpson's 1/3 rule is = 0.693314
The area value of 1/(1+x) by simpson's 3/8 rule is = 0.693398
PS C:\Users\Tonmo\OneDrive\Documents\RUET_CSE\CSE2204\Lab5\Q1>
```

Discussion: Numerical integration of the function $\frac{1}{1+x}$ using the Trapezoidal Rule, Simpson’s 1/3-Rule, and Simpson’s 3/8-Rule involves approximating the definite integral by dividing the interval into smaller segments. The Trapezoidal Rule calculates the area using trapezoids, while Simpson’s 1/3-Rule and 3/8-Rule utilize quadratic and cubic polynomial approximations, respectively. These methods improve accuracy compared to simple geometric shapes. The choice between them depends on the number of subintervals: Simpson’s 1/3-Rule requires an even number, while Simpson’s 3/8-Rule demands a multiple of 3. Adjusting the number of subintervals allows for a balance between computational efficiency and accuracy in estimating the integral.