**Problem No: 01**

**Problem Name:** Compute the values of using Trapezoidal rule, Simpson’s 1/3-rule and Simpson’s 3/8-rule, where h = 1/6.

**Objective:** To learn Trapezoidal rule, Simpson’s 1/3 rule and Simpson’s 3/8 rule.

**Theory:**

**Trapezoidal Rule:**

**Simpson’s 1/3-Rule:**

**Simpson’s 3/8-Rule:**

**Source Code:**

#include <iostream>

#include <iomanip>

#include <cmath>

using namespace std;

float x1 = 0.0, x2 = 1.0;

float h = 1.0 / 6.0;

int n = ceil(x2 / h);

float func(float x){return 1.0 / (1 + x);}

float trap(float \*x, float \*y);

float simp1(float \*x, float \*y);

float simp3(float \*x, float \*y);

int main()

{

float x[n + 1], y[n + 1];

int i;

for(i = 0; i < n + 1; i ++){

x[i] = x1;

y[i] = func(x1);

x1 += h;

}

cout << fixed;

cout << " x\t\t\t y\n";

for(i = 0; i < n + 1; i++)

cout << setprecision(6) << x[i] << "\t\t"<< setprecision(6) << y[i] << endl;

while(1){

int n1;

cout << "Press\t1. Trapizoidal\n\t2. Simpson's 1 / 3\n\t3. Simpson's 3 / 8\n\t4. exit\n";

cin >> n1;

if(n1 == 1)

cout << "Result: " << trap(x, y) << endl; //Trapizoidal Formula

else if(n1 == 2)

cout << "Result: " << simp1(x, y) << endl; //Simpson's 1 / 3 Formula

else if(n1 == 3)

cout << "Result: " << simp3(x, y) << endl; //Simpson's 3 / 8 Formula

else if(n1 == 4)

break;

}

return 0;

}

float trap(float \*x, float \*y) //Trapizoidal Formula

{

float sum = y[0] + y[n];

float s = 0;

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

s += y[i];

sum += (2 \* s);

sum \*= (h / 2);

return sum;

}

float simp1(float \*x, float \*y) //Simpson's 1 / 3 Formula

{

float sum = y[0] + y[n];

float s = 0;

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

if(i % 2 == 0)

s += (2 \* y[i]);

else if(i % 2 != 0)

s += (4 \* y[i]);

}

sum += s;

sum \*= (h / 3);

return sum;

}

float simp3(float \*x, float \*y) //Simpson's 3 / 8 Formula

{

float sum = y[0] + y[n];

float s = 0;

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

if(i % 3 == 0)

s += (2 \* y[i]);

else if(i % 3 != 0)

s += (3 \* y[i]);

}

sum += s;

sum \*= (3.0 \* h) / 8.0;

return sum;

}

**Output:**

