Code

//Preston Knibbe

//October 1, 2015

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

double Factorial(double number) {

double total = 1.0;

double other = number;

for (double i = 0.0; i < other; i++) {

total \*= number;

number--;

}

return total;

}

double MyExp(double x) {

double diff = 1.0;

double total = 1.0;

double counter = 1.0;

double temp = 1.0;

double power;

double fact;

// Using a difference between current an previous values will not produce accurate results

// when using whole number x's. Because of this MyExp function won't match the exp function in results.

// This could be fixed by comparing the values with the exp function values directly and continuing the

// loop until it is within a certain range of that number

while (abs(diff) >= 0.0001) {

power = pow(x, counter);

fact = Factorial(counter);

diff = temp - (power / fact);

temp = power / fact;

total += (power / fact);

counter++;

}

return total;

}

int main()

{

double x = 0;

cout << " X" << fixed << setw(12) << setprecision(4)

<< "MyExp()" << fixed << setw(12) << setprecision(4)

<< "Exp()" << fixed << setw(12) << setprecision(4) << endl;

for (x; x <= 10; x += 0.5) {

cout << x << fixed << setw(12) << setprecision(4)

<< MyExp(x) << fixed << setw(12) << setprecision(4)

<< exp(x) << fixed << setw(12) << setprecision(4) << endl;

}

return 0;

}

Code Output

X MyExp() Exp()

0.0000 1.0000 1.0000

0.5000 1.6487 1.6487

1.0000 2.0000 2.7183

1.5000 4.4817 4.4817

2.0000 5.0000 7.3891

2.5000 12.1825 12.1825

3.0000 13.0000 20.0855

3.5000 33.1154 33.1155

4.0000 34.3333 54.5982

4.5000 90.0171 90.0171

5.0000 91.4167 148.4132

5.5000 244.6919 244.6919

6.0000 244.6000 403.4288

6.5000 665.1416 665.1416

7.0000 656.5694 1096.6332

7.5000 1808.0424 1808.0424

8.0000 1766.3587 2980.9580

8.5000 4914.7688 4914.7688

9.0000 4759.8183 8103.0839

9.5000 13359.7268 13359.7268

10.0000 12842.3051 22026.4658

Process returned 0 (0x0) execution time : 0.070 s