Assignment 1

Name – Abhishek Agrahari

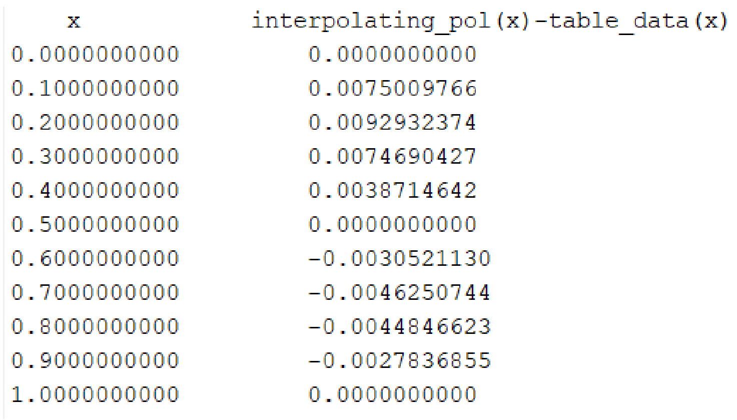
Roll Number – 190123066

Question 1-

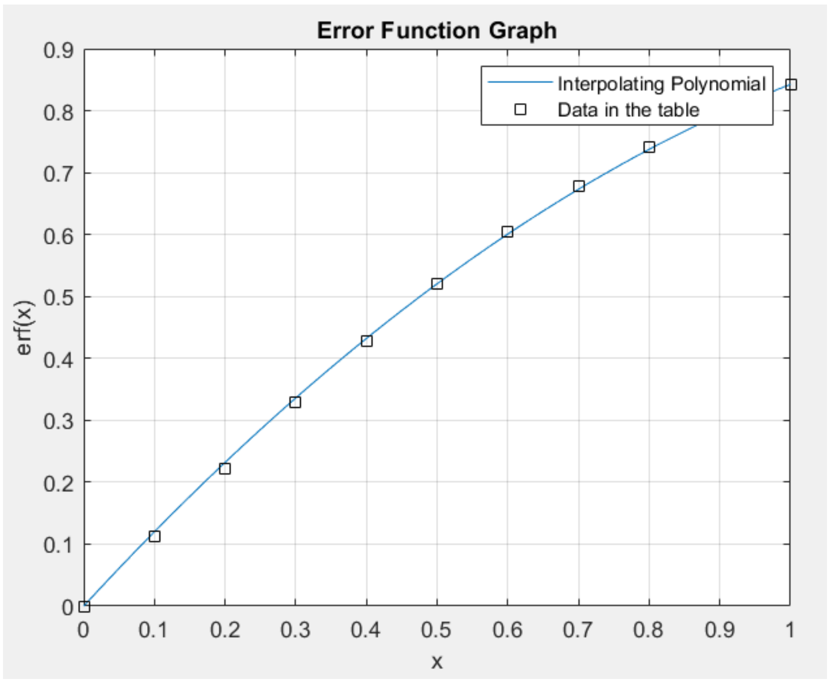
Code

```
clc;
clear;
n = 50;
x0 = linspace(0,1,n);
y0 = zeros(1,n);
for i = 1:n
    y0(i) = interpolating_polynomial(x0(i));
end
x1 = 0:0.1:1;
y1 = [0.00000000000000, 0.11246291601828, 0.22270258921048, 0.32862675945913, 0.42839235504667,
0.52049987781305, 0.60385609084793, 0.67780119383742, 0.74210096470766,
0.79690821242283,0.84270079294971];

fprintf("\tx \t\t\t interpolating_pol(x)-table_data(x)\n");
for t= 1:11
    fprintf("%.10f \t\t %.10f\n",x1(t),interpolating_polynomial(x1(t))-y1(t))
end

plot(x0,y0,x1,y1,'ks'), legend('Interpolating Polynomial', 'Data in the table'), xlabel('x'),
ylabel('erf(x)'), title('Error Function Graph'), grid on;

function [val] = interpolating_polynomial(t)
    x = [0, 0.5, 1.0];
    y_val = [0.00000000000000, 0.52049987781305, 0.84270079294971];
    L = @(x1,x2,x3) (((t-x2)*(t-x3))/((x1-x2)*(x1-x3)));
    val = y_val(1)*L(x(1),x(2),x(3)) + y_val(2)*L(x(2),x(1),x(3)) + y_val(3)*L(x(3),x(1),x(2));
end
```

Ouput



```
    x           interpolating_pol(x)-table_data(x)
0.0000000000        0.0000000000
0.1000000000        0.0075009766
0.2000000000        0.0092932374
0.3000000000        0.0074690427
0.4000000000        0.0038714642
0.5000000000        0.0000000000
0.6000000000        -0.0030521130
0.7000000000        -0.0046250744
0.8000000000        -0.0044846623
0.9000000000        -0.0027836855
1.0000000000        0.0000000000
```



Observations

As can be seen from the above table and the graph that the value of interpolating polynomial and data in the given table matches exactly at 0, 0.5 and 1. The value of interpolating polynomial is above the actual value for (0,0.5) and below the actual value for (0.5, 1). From the output it is evident that interpolating polynomial obtained is very accurate.

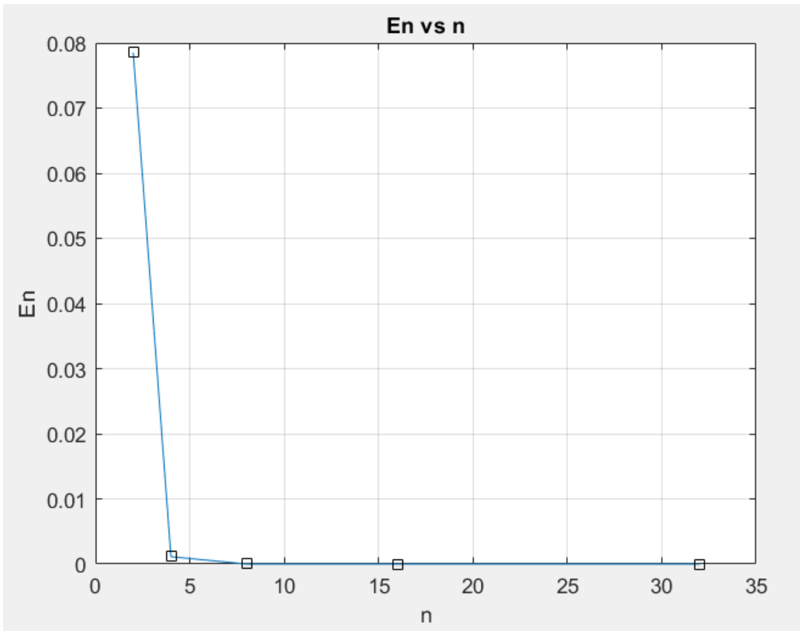
Question 2-

Code

```
clc;
clear;

fprintf("\tn \t\t\t En\n");
N = 5;
En = zeros(1,N);
X_axis = zeros(1,N);
for t = 1:N
    n = 2^t;
    nodal_pts = linspace(-1, 1, n + 1);
    dd = divided_difference_calculator(nodal_pts,n);
    En(t) = En_calculator(dd,nodal_pts,n+1);
    fprintf('\t%d\t%.15f\n', n, En(t));
    X_axis(t) = n;
end

plot(X_axis,En, X_axis,En, 'ks'), xlabel('n'), ylabel('En'), title('En vs n'), grid on;

function dd = divided_difference_calculator(nodal_pts,n)
    dd = zeros(n+1,n+1);
    for i = 1:n+1
        dd(i,1) = exp(nodal_pts(i));
    end
    for i = 2:n+1
        r = i-1;
        for c = 2:i
            dd(r,c) = (dd(r+1,c-1) - dd(r,c-1))/(nodal_pts(r+c-1)-nodal_pts(r));
            r = r - 1;
        end
    end
end

function En = En_calculator(dd, nodal_pts, n)
    x = linspace(-1,1,501);
    En = 0;
    for p = 1:501
        tk = x(p);
        val = 0;
        for i = 1:n
            term = 1;
            for j = 1:i-1
                term = term*(tk-nodal_pts(j));
            end
            val = val + term*dd(1,i);
        end
        En = max(En,abs(exp(tk) - val));
    end
end
```

Ouput

```
n                En
2    0.078525471129584
4    0.001124354920727
8    0.000000057999959
16   0.000000000000005
32   0.000000000400998
```



Observations

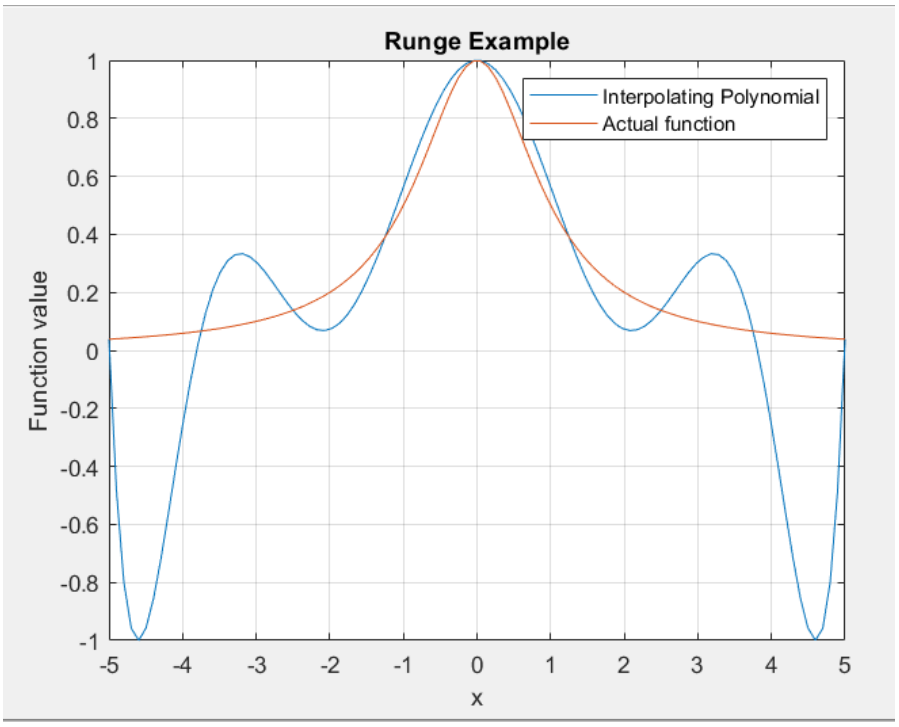
Value of En decreases initially as we take n from 2 to 16, but the En increases on moving from 16 to 32. This shows that taking more nodal points does not necessary going to decrease the error.

Question 3-

Code

```
clc;
clear;
n = 100;
x0 = linspace(-5,5,n);
y0 = zeros(1,n);
for i = 1:n
    y0(i) = interpolating_polynomial(x0(i));
end
y1 = 1./(1 + x0.^2);
plot(x0,y0,x0,y1), legend('Interpolating Polynomial', 'Actual function'), xlabel('x'),
ylabel('Function value'), title('Runge Example'), grid on;

function [val] = interpolating_polynomial(t)
    f = @(x) 1/(1 + (x*x));
    n = 9;
    x = linspace(-5,5,n);
    y = zeros(1,n);
    for i = 1:n
        y(i) = f(x(i));
    end
    val = 0.0;
    for i = 1:n
        sum = 1;
        for j = 1:n
            if j ~= i
                sum = sum * (t-x(j))/(x(i) - x(j));
            end
        end
        sum = sum * y(i);
        val = val + sum;
    end
end
```

Ouput



Observations

As can be seen from the output that even after taking 9 points, the graph of interpolating polynomial and the actual function are not coinciding well. This once again shows that taking more nodal points does not necessitates more accurate results.