```
%Question 1 on exam DONE
f = @(t, y) (exp(1)^(t-y));
a = 0;
b = 1;
yo = 1;
N = 10;
realF = @(t) (log(exp(1)^t + exp(1) - 1));
approx = Taylors(f, a, b, N, yo, 4, realF);
gError1 = [abs(approx(3, 1:end) - approx(2, 1:end))];
maxG1 = max(gError1);
maxg1 = gError1(length(gError1));
for i =1:length(approx)
   fprintf("t = %.2f w = %.10f real value is %.10f\n", approx(1, i),
 approx(2, i), approx(3,i));
end

N = 20;
realF = @(t) (log(exp(1)^t + exp(1) - 1));
approx = Taylors(f, a, b, N, yo, 4, realF);
gError2 = [abs(approx(3, 1:end) - approx(2, 1:end))];
maxG2 = max(gError2);
maxg2 = gError2(length(gError2));

N = 40;
realF = @(t) (log(exp(1)^t + exp(1) - 1));
approx = Taylors(f, a, b, N, yo, 4, realF);
gError3 = [abs(approx(3, 1:end) - approx(2, 1:end))];
maxG3 = max(gError3);
maxg3 = gError3(length(gError3));


N = 80;
realF = @(t) (log(exp(1)^t + exp(1) - 1));
approx = Taylors(f, a, b, N, yo, 4, realF);
gError4 = [abs(approx(3, 1:end) - approx(2, 1:end))];
maxG4 = max(gError4);
maxg4 = gError4(length(gError4));


N = 160;
realF = @(t) (log(exp(1)^t + exp(1) - 1));
approx = Taylors(f, a, b, N, yo, 4, realF);
gError5 = [abs(approx(3, 1:end) - approx(2, 1:end))];
maxG5 = max(gError5);
maxg5 = gError5(length(gError5));

fprintf("N = 10 Max global error = %.15f\n", maxG1);
fprintf("N = 20 Max global error = %.15f\n", maxG2);
fprintf("N = 40 Max global error = %.15f\n", maxG3);
fprintf("N = 80 Max global error = %.15f\n", maxG4);
fprintf("N = 160 Max global error = %.15f\n", maxG5);
```

```
fprintf("The maximum global error seems to be getting 1/16th'd every
 time we double our step size\n");
fprintf("This is exactly what we should expect since the Taylor method
 order four has O(h^4) error\n");
fprintf("So as our step size is cut in half, or as h->h/2, the error
 is reduced by 1/2^4 = 1/16\n");
```

```
t = 0.10 w = 1.0379605232 real value is 1.0379605142
t = 0.20 w = 1.0783023086 real value is 1.0783022921
t = 0.30 w = 1.1210717441 real value is 1.1210717221
t = 0.40 w = 1.1663041478 real value is 1.1663041223
t = 0.50 w = 1.2140230876 real value is 1.2140230606
t = 0.60 w = 1.2642399189 real value is 1.2642398927
t = 0.70 w = 1.3169535602 real value is 1.3169535369
t = 0.80 w = 1.3721505142 real value is 1.3721504958
t = 0.90 w = 1.4298051351 real value is 1.4298051232
t = 1.00 w = 1.4898801298 real value is 1.4898801256
N = 10 Max global error = 0.000000026929541
N = 20 Max global error = 0.000000001585931
N = 40 Max global error = 0.000000000096136
N = 80 Max global error = 0.000000000005919
N = 160 Max global error = 0.000000000000367
The maximum global error seems to be getting 1/16th'd every time we
 double our step size
This is exactly what we should expect since the Taylor method order
 four has O(h^4) error
So as our step size is cut in half, or as h->h/2, the error is reduced
 by 1/2^4 = 1/16
```

Published with MATLAB® R2018a