Michael Nguyen
mnguy181@ucsc.edu
1/24/2021

CSE13S Winter 2021
Assignment 2: A Small Numerical Library
Writeup Document

**Note: The following includes output from running mathlib-test.c for each function and also provides some graphs to showcase the differences in output between my function and the expected output from the functions from the <math.h> library. Although in the lab document, it presented a print statement, I temporarily edited the print statement to allow for more decimal places to ensure that the viewer can see that there are indeed differences in output.

OUTPUT for Sin(x)

The output for running sin(x) in the specified range [-2pi, 2pi) as shown in terminal is as follows.

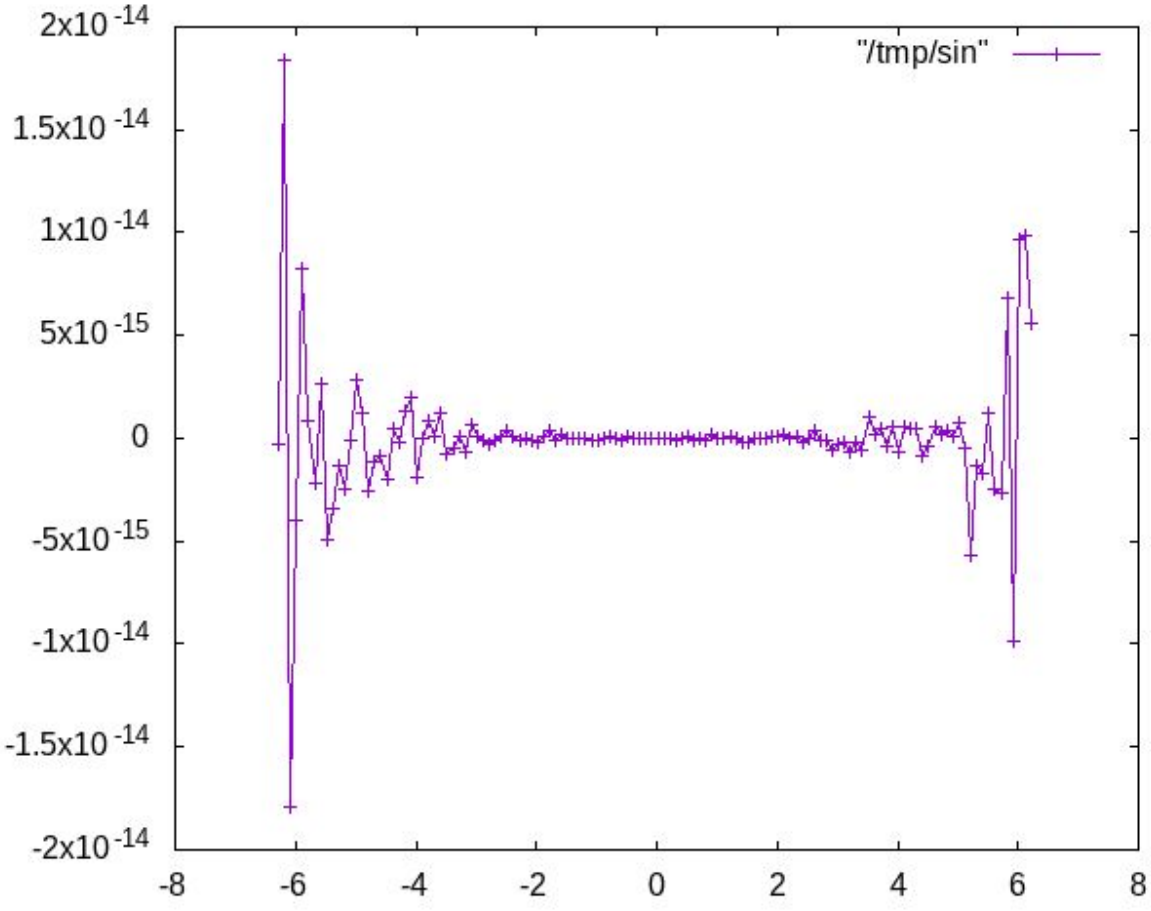
x	Sin	Library	Difference
-6.2832	-0.00000000	0.00000000	-0.00000000000000024493
-6.1832	0.09983342	0.09983342	0.00000000000001842970
-6.0832	0.19866933	0.19866933	-0.00000000000001790235
-5.9832	0.29552021	0.29552021	-0.00000000000000394129
-5.8832	0.38941834	0.38941834	0.00000000000000827116
-5.7832	0.47942554	0.47942554	0.00000000000000088818
-5.6832	0.56464247	0.56464247	-0.00000000000000222045
-5.5832	0.64421769	0.64421769	0.00000000000000266454
-5.4832	0.71735609	0.71735609	-0.00000000000000488498
-5.3832	0.78332691	0.78332691	-0.00000000000000344169
-5.2832	0.84147098	0.84147098	-0.00000000000000133227
-5.1832	0.89120736	0.89120736	-0.00000000000000244249
-5.0832	0.93203909	0.93203909	-0.00000000000000011102
-4.9832	0.96355819	0.96355819	0.00000000000000288658
-4.8832	0.98544973	0.98544973	0.00000000000000122125
-4.7832	0.99749499	0.99749499	-0.00000000000000255351
-4.6832	0.99957360	0.99957360	-0.00000000000000111022
-4.5832	0.99166481	0.99166481	-0.00000000000000088818
-4.4832	0.97384763	0.97384763	-0.00000000000000199840
-4.3832	0.94630009	0.94630009	0.00000000000000044409
-4.2832	0.90929743	0.90929743	-0.00000000000000022204
-4.1832	0.86320937	0.86320937	0.00000000000000133227
-4.0832	0.80849640	0.80849640	0.00000000000000199840
-3.9832	0.74570521	0.74570521	-0.00000000000000188738
-3.8832	0.67546318	0.67546318	0.00000000000000000000
-3.7832	0.59847214	0.59847214	0.00000000000000088818

-3.6832	0.51550137	0.51550137	0.00000000000000011102
-3.5832	0.42737988	0.42737988	0.00000000000000127676
-3.4832	0.33498815	0.33498815	-0.00000000000000072164
-3.3832	0.23924933	0.23924933	-0.00000000000000044409
-3.2832	0.14112001	0.14112001	0.00000000000000008327
-3.1832	0.04158066	0.04158066	-0.00000000000000062450
-3.0832	-0.05837414	-0.05837414	0.00000000000000063838
-2.9832	-0.15774569	-0.15774569	0.00000000000000013878
-2.8832	-0.25554110	-0.25554110	-0.00000000000000005551
-2.7832	-0.35078323	-0.35078323	-0.00000000000000027756
-2.6832	-0.44252044	-0.44252044	-0.00000000000000005551
-2.5832	-0.52983614	-0.52983614	0.00000000000000011102
-2.4832	-0.61185789	-0.61185789	0.00000000000000033307
-2.3832	-0.68776616	-0.68776616	0.00000000000000011102
-2.2832	-0.75680250	-0.75680250	-0.00000000000000011102
-2.1832	-0.81827711	-0.81827711	0.00000000000000000000
-2.0832	-0.87157577	-0.87157577	-0.00000000000000011102
-1.9832	-0.91616594	-0.91616594	-0.00000000000000022204
-1.8832	-0.95160207	-0.95160207	0.00000000000000011102
-1.7832	-0.97753012	-0.97753012	0.00000000000000033307
-1.6832	-0.99369100	-0.99369100	-0.00000000000000011102
-1.5832	-0.99992326	-0.99992326	0.00000000000000022204
-1.4832	-0.99616461	-0.99616461	0.00000000000000000000
-1.3832	-0.98245261	-0.98245261	0.00000000000000000000
-1.2832	-0.95892427	-0.95892427	0.00000000000000000000
-1.1832	-0.92581468	-0.92581468	0.00000000000000000000
-1.0832	-0.88345466	-0.88345466	-0.00000000000000011102
-0.9832	-0.83226744	-0.83226744	-0.00000000000000011102
-0.8832	-0.77276449	-0.77276449	0.00000000000000000000
-0.7832	-0.70554033	-0.70554033	0.00000000000000011102
-0.6832	-0.63126664	-0.63126664	0.00000000000000000000
-0.5832	-0.55068554	-0.55068554	-0.00000000000000011102
-0.4832	-0.46460218	-0.46460218	0.00000000000000005551
-0.3832	-0.37387666	-0.37387666	0.00000000000000000000
-0.2832	-0.27941550	-0.27941550	0.00000000000000000000
-0.1832	-0.18216250	-0.18216250	0.00000000000000002776
-0.0832	-0.08308940	-0.08308940	0.00000000000000000000
0.0168	0.01681390	0.01681390	0.00000000000000000000
0.1168	0.11654920	0.11654920	0.00000000000000000000
0.2168	0.21511999	0.21511999	0.00000000000000000000
0.3168	0.31154136	0.31154136	-0.00000000000000005551
0.4168	0.40484992	0.40484992	0.00000000000000000000
0.5168	0.49411335	0.49411335	0.00000000000000005551
0.6168	0.57843976	0.57843976	-0.00000000000000011102
0.7168	0.65698660	0.65698660	0.00000000000000000000

0.8168	0.72896904	0.72896904	-0.00000000000000011102
0.9168	0.79366786	0.79366786	0.00000000000000022204
1.0168	0.85043662	0.85043662	0.00000000000000000000
1.1168	0.89870810	0.89870810	0.00000000000000000000
1.2168	0.93799998	0.93799998	0.00000000000000011102
1.3168	0.96791967	0.96791967	0.00000000000000000000
1.4168	0.98816823	0.98816823	-0.00000000000000022204
1.5168	0.99854335	0.99854335	-0.00000000000000022204
1.6168	0.99894134	0.99894134	0.00000000000000000000
1.7168	0.98935825	0.98935825	0.00000000000000000000
1.8168	0.96988981	0.96988981	0.00000000000000000000
1.9168	0.94073056	0.94073056	0.00000000000000011102
2.0168	0.90217183	0.90217183	0.00000000000000011102
2.1168	0.85459891	0.85459891	0.00000000000000022204
2.2168	0.79848711	0.79848711	0.00000000000000000000
2.3168	0.73439710	0.73439710	0.00000000000000011102
2.4168	0.66296923	0.66296923	-0.00000000000000022204
2.5168	0.58491719	0.58491719	0.00000000000000000000
2.6168	0.50102086	0.50102086	0.00000000000000033307
2.7168	0.41211849	0.41211849	-0.00000000000000005551
2.8168	0.31909836	0.31909836	-0.00000000000000011102
2.9168	0.22288991	0.22288991	-0.00000000000000058287
3.0168	0.12445442	0.12445442	-0.00000000000000029143
3.1168	0.02477543	0.02477543	-0.00000000000000022551
3.2168	-0.07515112	-0.07515112	-0.00000000000000069389
3.3168	-0.17432678	-0.17432678	-0.00000000000000019429
3.4168	-0.27176063	-0.27176063	-0.00000000000000061062
3.5168	-0.36647913	-0.36647913	0.00000000000000099920
3.6168	-0.45753589	-0.45753589	0.00000000000000016653
3.7168	-0.54402111	-0.54402111	0.00000000000000044409
3.8168	-0.62507065	-0.62507065	-0.00000000000000033307
3.9168	-0.69987469	-0.69987469	0.00000000000000055511
4.0168	-0.76768581	-0.76768581	-0.00000000000000066613
4.1168	-0.82782647	-0.82782647	0.00000000000000055511
4.2168	-0.87969576	-0.87969576	0.00000000000000044409
4.3168	-0.92277542	-0.92277542	0.00000000000000044409
4.4168	-0.95663502	-0.95663502	-0.00000000000000088818
4.5168	-0.98093623	-0.98093623	-0.00000000000000033307
4.6168	-0.99543625	-0.99543625	0.00000000000000055511
4.7168	-0.99999021	-0.99999021	0.00000000000000022204
4.8168	-0.99455259	-0.99455259	0.00000000000000033307
4.9168	-0.97917773	-0.97917773	0.00000000000000011102
5.0168	-0.95401925	-0.95401925	0.00000000000000077716
5.1168	-0.91932853	-0.91932853	-0.00000000000000044409
5.2168	-0.87545217	-0.87545217	-0.00000000000000566214

5.3168	-0.82282859	-0.82282859	-0.00000000000000133227
5.4168	-0.76198358	-0.76198358	-0.00000000000000166533
5.5168	-0.69352508	-0.69352508	0.00000000000000122125
5.6168	-0.61813711	-0.61813711	-0.00000000000000244249
5.7168	-0.53657292	-0.53657292	-0.00000000000000266454
5.8168	-0.44964746	-0.44964746	0.00000000000000682787
5.9168	-0.35822928	-0.35822928	-0.00000000000000982547
6.0168	-0.26323179	-0.26323179	0.00000000000000965894
6.1168	-0.16560418	-0.16560418	0.00000000000000982547
6.2168	-0.06632190	-0.06632190	0.00000000000000556499

Here is a graph of the differences.



Analysis:
As seen by the graph, the differences appear to be larger as the value x distances itself from the center 0. In other words, as x approaches the boundaries/range set (as [-2pi, 2pi), the differences between my function and math.h function of sin(x) increases. This is likely due to the number of terms that are needed to compute the total approximation. Thus, it makes sense that as the value of x approaches the boundaries -2pi and 2pi, the approximation that we computed becomes less accurate (since more terms may be needed to yield an accurate approximation), yielding larger differences between my function and the math.h library function of sin(x).

OUTPUT for Cos(x)

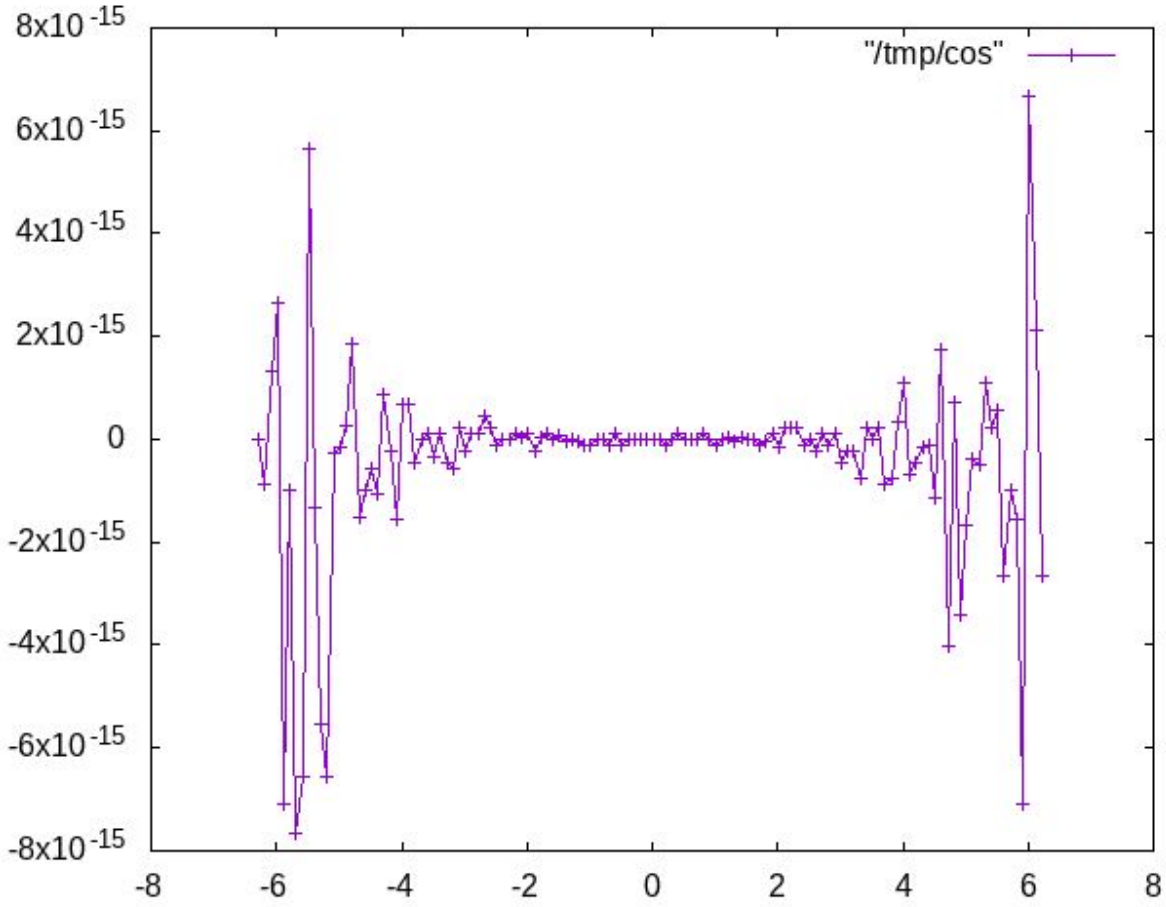
The output for running Cos(x) in the specified range [-2pi, 2pi) as shown in terminal is as follows.

x	Cos	Library	Difference
-6.2832	1.00000000	1.00000000	0.00000000000000000000
-6.1832	0.99500417	0.99500417	-0.00000000000000088818
-6.0832	0.98006658	0.98006658	0.00000000000000133227
-5.9832	0.95533649	0.95533649	0.00000000000000266454
-5.8832	0.92106099	0.92106099	-0.00000000000000710543
-5.7832	0.87758256	0.87758256	-0.00000000000000099920
-5.6832	0.82533561	0.82533561	-0.00000000000000766054
-5.5832	0.76484219	0.76484219	-0.00000000000000655032
-5.4832	0.69670671	0.69670671	0.00000000000000566214
-5.3832	0.62160997	0.62160997	-0.00000000000000133227
-5.2832	0.54030231	0.54030231	-0.00000000000000555112
-5.1832	0.45359612	0.45359612	-0.00000000000000655032
-5.0832	0.36235775	0.36235775	-0.00000000000000027756
-4.9832	0.26749883	0.26749883	-0.00000000000000016653
-4.8832	0.16996714	0.16996714	0.00000000000000027756
-4.7832	0.07073720	0.07073720	0.00000000000000187350
-4.6832	-0.02919952	-0.02919952	-0.00000000000000152309
-4.5832	-0.12884449	-0.12884449	-0.00000000000000099920
-4.4832	-0.22720209	-0.22720209	-0.00000000000000058287
-4.3832	-0.32328957	-0.32328957	-0.00000000000000105471
-4.2832	-0.41614684	-0.41614684	0.00000000000000088818
-4.1832	-0.50484610	-0.50484610	-0.00000000000000022204
-4.0832	-0.58850112	-0.58850112	-0.00000000000000155431
-3.9832	-0.66627602	-0.66627602	0.00000000000000066613
-3.8832	-0.73739372	-0.73739372	0.00000000000000066613
-3.7832	-0.80114362	-0.80114362	-0.00000000000000044409
-3.6832	-0.85688875	-0.85688875	0.00000000000000000000
-3.5832	-0.90407214	-0.90407214	0.00000000000000011102
-3.4832	-0.94222234	-0.94222234	-0.00000000000000033307
-3.3832	-0.97095817	-0.97095817	0.00000000000000011102
-3.2832	-0.98999250	-0.98999250	-0.00000000000000044409
-3.1832	-0.99913515	-0.99913515	-0.00000000000000055511
-3.0832	-0.99829478	-0.99829478	0.00000000000000022204
-2.9832	-0.98747977	-0.98747977	-0.00000000000000022204
-2.8832	-0.96679819	-0.96679819	0.00000000000000011102
-2.7832	-0.93645669	-0.93645669	0.00000000000000011102
-2.6832	-0.89675842	-0.89675842	0.00000000000000044409
-2.5832	-0.84810003	-0.84810003	0.00000000000000022204
-2.4832	-0.79096771	-0.79096771	-0.00000000000000011102

-2.3832	-0.72593230	-0.72593230	0.00000000000000000000
-2.2832	-0.65364362	-0.65364362	0.00000000000000000000
-2.1832	-0.57482395	-0.57482395	0.00000000000000011102
-2.0832	-0.49026082	-0.49026082	0.00000000000000005551
-1.9832	-0.40079917	-0.40079917	0.00000000000000011102
-1.8832	-0.30733287	-0.30733287	-0.00000000000000022204
-1.7832	-0.21079580	-0.21079580	0.00000000000000005551
-1.6832	-0.11215253	-0.11215253	0.00000000000000009714
-1.5832	-0.01238866	-0.01238866	0.00000000000000001388
-1.4832	0.08749898	0.08749898	0.00000000000000008327
-1.3832	0.18651237	0.18651237	-0.00000000000000002776
-1.2832	0.28366219	0.28366219	0.00000000000000000000
-1.1832	0.37797774	0.37797774	-0.00000000000000005551
-1.0832	0.46851667	0.46851667	-0.00000000000000011102
-0.9832	0.55437434	0.55437434	-0.00000000000000011102
-0.8832	0.63469288	0.63469288	0.00000000000000000000
-0.7832	0.70866977	0.70866977	0.00000000000000000000
-0.6832	0.77556588	0.77556588	-0.00000000000000011102
-0.5832	0.83471278	0.83471278	0.00000000000000011102
-0.4832	0.88551952	0.88551952	-0.00000000000000011102
-0.3832	0.92747843	0.92747843	0.00000000000000000000
-0.2832	0.96017029	0.96017029	0.00000000000000000000
-0.1832	0.98326844	0.98326844	0.00000000000000000000
-0.0832	0.99654210	0.99654210	0.00000000000000000000
0.0168	0.99985864	0.99985864	0.00000000000000000000
0.1168	0.99318492	0.99318492	0.00000000000000000000
0.2168	0.97658763	0.97658763	-0.00000000000000011102
0.3168	0.95023259	0.95023259	0.00000000000000000000
0.4168	0.91438315	0.91438315	0.00000000000000011102
0.5168	0.86939749	0.86939749	0.00000000000000000000
0.6168	0.81572510	0.81572510	0.00000000000000000000
0.7168	0.75390225	0.75390225	0.00000000000000000000
0.8168	0.68454667	0.68454667	0.00000000000000011102
0.9168	0.60835131	0.60835131	0.00000000000000000000
1.0168	0.52607752	0.52607752	-0.00000000000000011102
1.1168	0.43854733	0.43854733	0.00000000000000000000
1.2168	0.34663532	0.34663532	0.00000000000000005551
1.3168	0.25125984	0.25125984	-0.00000000000000005551
1.4168	0.15337386	0.15337386	0.00000000000000002776
1.5168	0.05395542	0.05395542	0.00000000000000000000
1.6168	-0.04600213	-0.04600213	0.00000000000000001388
1.7168	-0.14550003	-0.14550003	-0.00000000000000011102
1.8168	-0.24354415	-0.24354415	-0.00000000000000002776
1.9168	-0.33915486	-0.33915486	0.00000000000000011102
2.0168	-0.43137684	-0.43137684	-0.00000000000000016653

2.1168	-0.51928865	-0.51928865	0.00000000000000022204
2.2168	-0.60201190	-0.60201190	0.00000000000000022204
2.3168	-0.67872005	-0.67872005	0.00000000000000022204
2.4168	-0.74864665	-0.74864665	-0.00000000000000011102
2.5168	-0.81109301	-0.81109301	0.00000000000000000000
2.6168	-0.86543521	-0.86543521	-0.00000000000000022204
2.7168	-0.91113026	-0.91113026	0.00000000000000011102
2.8168	-0.94772160	-0.94772160	-0.00000000000000011102
2.9168	-0.97484362	-0.97484362	0.00000000000000011102
3.0168	-0.99222533	-0.99222533	-0.00000000000000044409
3.1168	-0.99969304	-0.99969304	-0.00000000000000022204
3.2168	-0.99717216	-0.99717216	-0.00000000000000022204
3.3168	-0.98468786	-0.98468786	-0.00000000000000077716
3.4168	-0.96236488	-0.96236488	0.00000000000000022204
3.5168	-0.93042627	-0.93042627	0.00000000000000000000
3.6168	-0.88919115	-0.88919115	0.00000000000000022204
3.7168	-0.83907153	-0.83907153	-0.00000000000000088818
3.8168	-0.78056818	-0.78056818	-0.00000000000000077716
3.9168	-0.71426565	-0.71426565	0.00000000000000033307
4.0168	-0.64082642	-0.64082642	0.00000000000000111022
4.1168	-0.56098426	-0.56098426	-0.00000000000000066613
4.2168	-0.47553693	-0.47553693	-0.00000000000000044409
4.3168	-0.38533819	-0.38533819	-0.00000000000000016653
4.4168	-0.29128928	-0.29128928	-0.00000000000000011102
4.5168	-0.19432991	-0.19432991	-0.00000000000000113798
4.6168	-0.09542885	-0.09542885	0.00000000000000176248
4.7168	0.00442570	0.00442570	-0.00000000000000402976
4.8168	0.10423603	0.10423603	0.00000000000000073552
4.9168	0.20300486	0.20300486	-0.00000000000000341394
5.0168	0.29974534	0.29974534	-0.00000000000000166533
5.1168	0.39349087	0.39349087	-0.00000000000000038858
5.2168	0.48330476	0.48330476	-0.00000000000000049960
5.3168	0.56828963	0.56828963	0.00000000000000111022
5.4168	0.64759634	0.64759634	0.00000000000000022204
5.5168	0.72043248	0.72043248	0.00000000000000055511
5.6168	0.78607030	0.78607030	-0.00000000000000266454
5.7168	0.84385396	0.84385396	-0.00000000000000099920
5.8168	0.89320611	0.89320611	-0.00000000000000155431
5.9168	0.93363364	0.93363364	-0.00000000000000710543
6.0168	0.96473262	0.96473262	0.00000000000000666134
6.1168	0.98619230	0.98619230	0.00000000000000210942
6.2168	0.99779828	0.99779828	-0.00000000000000266454

Here is a graph of the differences.



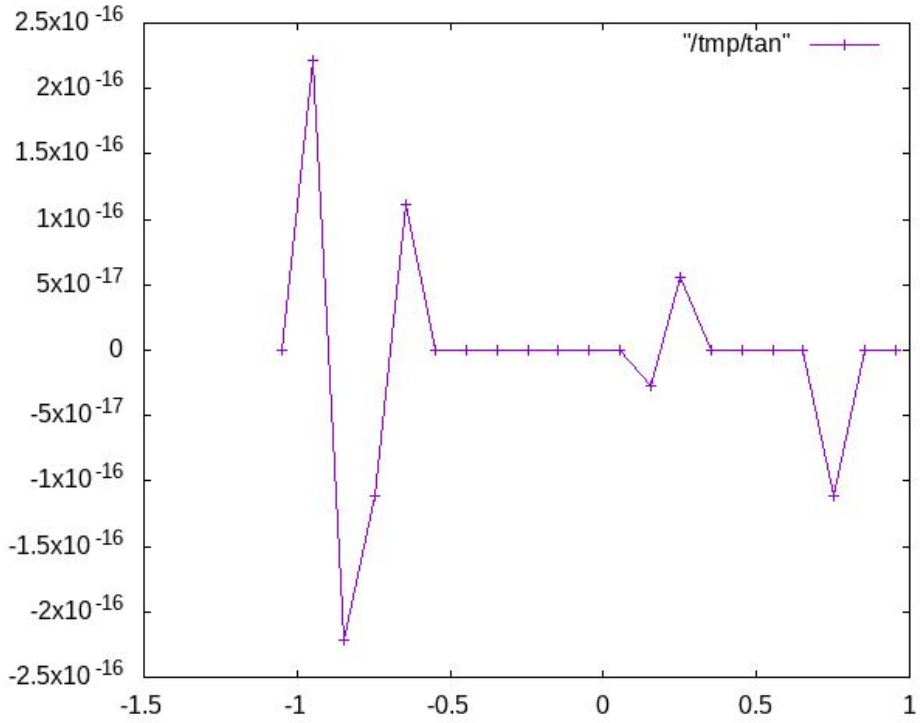
Analysis:
As seen by the graph and from my output, the differences appear to be larger as the value x distances itself from the center 0. In other words, as x approaches the boundaries/range set (as [-2pi, 2pi), the differences between my function and math.h function of cos(x) increases. This is very similar to the data that we found for sin(x) and it is likely due to the number of terms that are needed to compute the total approximation. When approaching these boundaries, the approximation in the function that I implemented using the Taylor Expansion method becomes less and less accurate since there may be a need for calculating more terms. Thus, it makes sense that as the value of x approaches the boundaries -2pi and 2pi, the approximation that we computed becomes less accurate (since there is a need for an even greater amount of terms to yield an accurate approximation), yielding larger differences between my function and the math.h library function of cos(x).

OUTPUT for Tan(x)
The output for running Tan(x) in the specified range [-pi/3, pi/3) as shown in terminal is as follows.

x	Tan	Library	Difference
-1.0472	-1.73205081	-1.73205081	0.00000000000000000000
-0.9472	-1.39013234	-1.39013234	0.00000000000000022204

-0.8472	-1.13191929	-1.13191929	-0.00000000000000022204
-0.7472	-0.92637546	-0.92637546	-0.00000000000000011102
-0.6472	-0.75579177	-0.75579177	0.00000000000000011102
-0.5472	-0.60925593	-0.60925593	0.00000000000000000000
-0.4472	-0.47960335	-0.47960335	0.00000000000000000000
-0.3472	-0.36185587	-0.36185587	0.00000000000000000000
-0.2472	-0.25235888	-0.25235888	0.00000000000000000000
-0.1472	-0.14826996	-0.14826996	0.00000000000000000000
-0.0472	-0.04723263	-0.04723263	0.00000000000000000000
0.0528	0.05285158	0.05285158	0.00000000000000000000
0.1528	0.15400290	0.15400290	-0.00000000000000002776
0.2528	0.25832923	0.25832923	0.00000000000000005551
0.3528	0.36820762	0.36820762	0.00000000000000000000
0.4528	0.48651614	0.48651614	0.00000000000000000000
0.5528	0.61696774	0.61696774	0.00000000000000000000
0.6528	0.76463587	0.76463587	0.00000000000000000000
0.7528	0.93684479	0.93684479	-0.00000000000000011102
0.8528	1.14478719	1.14478719	0.00000000000000000000
0.9528	1.40669776	1.40669776	0.00000000000000000000

Here is a graph of the differences.



Analysis:
As seen by the graph and from my output, the differences appear to be larger as the value x distances itself from the center 0. In other words, as x approaches the boundaries/range set (as [-pi/3, pi/3), the differences between my function and math.h function of tan(x) increases. Again,

this is very similar to the cases of sin(x) and cos(x) in which more terms may be needed to form an accurate approximation. In addition, these differences between the two functions could also be due to implementation. In my function, I relied on sin(x) and cos(x) which both utilized Taylor Expansion methods to find the approximation for sin(x)/cos(x). The implementation for the <math.h> library tan(x) function could have been done differently and more accurately utilizing more terms, thus forming differences as the value of x distances itself from 0.

OUTPUT of Exp(x)

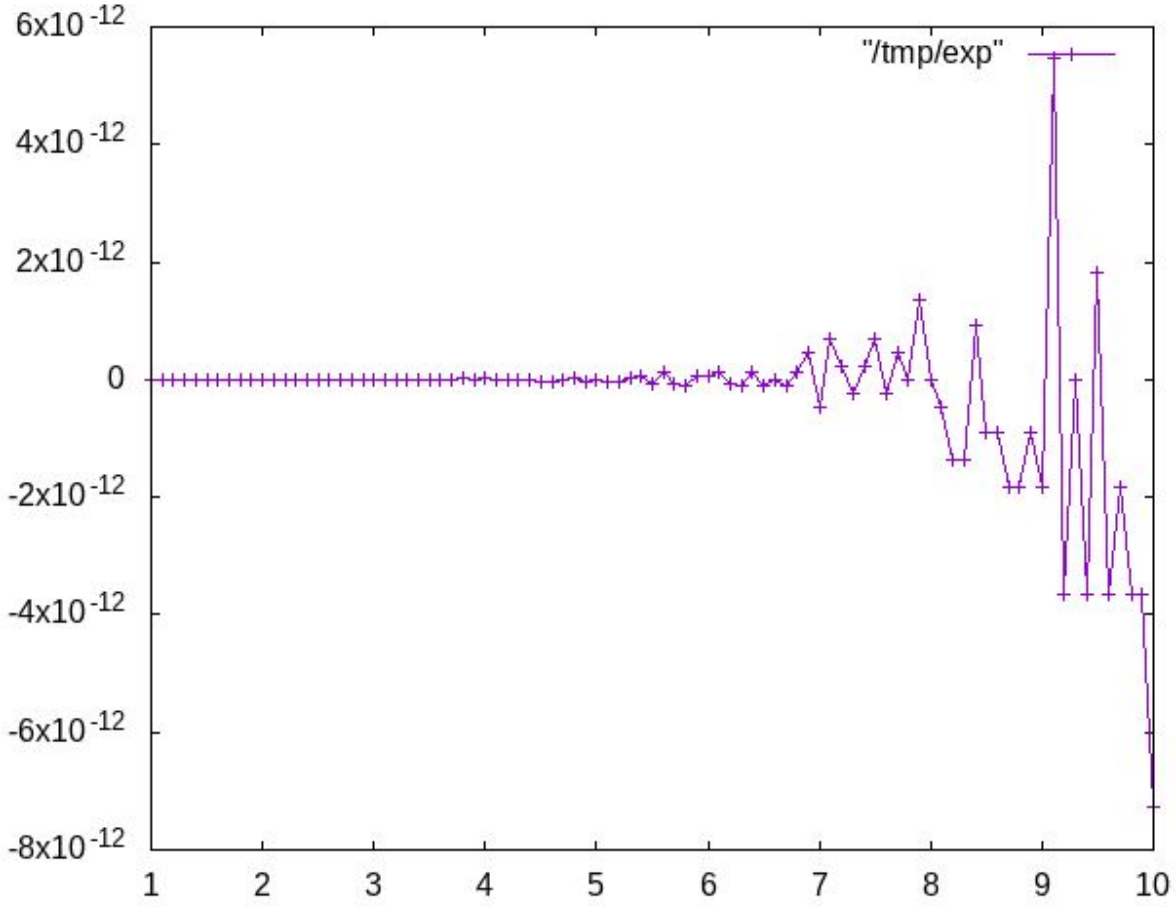
The output for running Exp(x) in the specified range [1, 10) as shown in terminal is as follows.

x	Exp	Library	Difference
1.0000	2.71828183	2.71828183	0.00000000000000044409
1.1000	3.00416602	3.00416602	0.00000000000000044409
1.2000	3.32011692	3.32011692	0.00000000000000000000
1.3000	3.66929667	3.66929667	-0.00000000000000044409
1.4000	4.05519997	4.05519997	-0.00000000000000088818
1.5000	4.48168907	4.48168907	0.00000000000000088818
1.6000	4.95303242	4.95303242	-0.00000000000000088818
1.7000	5.47394739	5.47394739	-0.00000000000000088818
1.8000	6.04964746	6.04964746	-0.00000000000000177636
1.9000	6.68589444	6.68589444	0.00000000000000000000
2.0000	7.38905610	7.38905610	-0.00000000000000088818
2.1000	8.16616991	8.16616991	0.00000000000000000000
2.2000	9.02501350	9.02501350	-0.00000000000000532907
2.3000	9.97418245	9.97418245	-0.00000000000000177636
2.4000	11.02317638	11.02317638	0.00000000000000000000
2.5000	12.18249396	12.18249396	-0.00000000000000177636
2.6000	13.46373804	13.46373804	0.00000000000000000000
2.7000	14.87973172	14.87973172	0.00000000000000000000
2.8000	16.44464677	16.44464677	-0.00000000000000710543
2.9000	18.17414537	18.17414537	0.00000000000000000000
3.0000	20.08553692	20.08553692	-0.00000000000000710543
3.1000	22.19795128	22.19795128	-0.00000000000000710543
3.2000	24.53253020	24.53253020	-0.00000000000000355271
3.3000	27.11263892	27.11263892	0.00000000000000355271
3.4000	29.96410005	29.96410005	0.00000000000000000000
3.5000	33.11545196	33.11545196	0.00000000000000000000
3.6000	36.59823444	36.59823444	0.00000000000000710543
3.7000	40.44730436	40.44730436	0.00000000000000000000
3.8000	44.70118449	44.70118449	0.00000000000001421085
3.9000	49.40244911	49.40244911	0.00000000000000710543
4.0000	54.59815003	54.59815003	0.00000000000002131628
4.1000	60.34028760	60.34028760	-0.00000000000000710543
4.2000	66.68633104	66.68633104	-0.00000000000001421085

4.3000	73.69979370	73.69979370	-0.00000000000001421085
4.4000	81.45086866	81.45086866	-0.00000000000001421085
4.5000	90.01713130	90.01713130	-0.00000000000004263256
4.6000	99.48431564	99.48431564	-0.00000000000002842171
4.7000	109.94717245	109.94717245	0.00000000000000000000
4.8000	121.51041752	121.51041752	0.00000000000004263256
4.9000	134.28977968	134.28977968	-0.00000000000002842171
5.0000	148.41315910	148.41315910	0.00000000000000000000
5.1000	164.02190730	164.02190730	-0.00000000000002842171
5.2000	181.27224188	181.27224188	-0.00000000000002842171
5.3000	200.33680997	200.33680997	0.00000000000002842171
5.4000	221.40641620	221.40641620	0.00000000000005684342
5.5000	244.69193226	244.69193226	-0.00000000000005684342
5.6000	270.42640743	270.42640743	0.00000000000011368684
5.7000	298.86740097	298.86740097	-0.00000000000005684342
5.8000	330.29955991	330.29955991	-0.00000000000011368684
5.9000	365.03746787	365.03746787	0.00000000000005684342
6.0000	403.42879349	403.42879349	0.00000000000005684342
6.1000	445.85777008	445.85777008	0.00000000000011368684
6.2000	492.74904109	492.74904109	-0.00000000000005684342
6.3000	544.57191013	544.57191013	-0.00000000000011368684
6.4000	601.84503787	601.84503787	0.00000000000011368684
6.5000	665.14163304	665.14163304	-0.00000000000011368684
6.6000	735.09518924	735.09518924	0.00000000000000000000
6.7000	812.40582517	812.40582517	-0.00000000000011368684
6.8000	897.84729165	897.84729165	0.00000000000011368684
6.9000	992.27471561	992.27471561	0.00000000000045474735
7.0000	1096.63315843	1096.63315843	-0.00000000000045474735
7.1000	1211.96707449	1211.96707449	0.00000000000068212103
7.2000	1339.43076439	1339.43076439	0.00000000000022737368
7.3000	1480.29992758	1480.29992758	-0.00000000000022737368
7.4000	1635.98443000	1635.98443000	0.00000000000022737368
7.5000	1808.04241446	1808.04241446	0.00000000000068212103
7.6000	1998.19589510	1998.19589510	-0.00000000000022737368
7.7000	2208.34799189	2208.34799189	0.00000000000045474735
7.8000	2440.60197762	2440.60197762	0.00000000000000000000
7.9000	2697.28232827	2697.28232827	0.00000000000136424205
8.0000	2980.95798704	2980.95798704	0.00000000000000000000
8.1000	3294.46807528	3294.46807528	-0.00000000000045474735
8.2000	3640.95030733	3640.95030733	-0.00000000000136424205
8.3000	4023.87239382	4023.87239382	-0.00000000000136424205
8.4000	4447.06674770	4447.06674770	0.00000000000090949470
8.5000	4914.76884030	4914.76884030	-0.00000000000090949470
8.6000	5431.65959136	5431.65959136	-0.00000000000090949470
8.7000	6002.91221726	6002.91221726	-0.00000000000181898940

8.8000	6634.24400628	6634.24400628	-0.00000000000181898940
8.9000	7331.97353916	7331.97353916	-0.00000000000090949470
9.0000	8103.08392758	8103.08392758	-0.00000000000181898940
9.1000	8955.29270348	8955.29270348	0.00000000000545696821
9.2000	9897.12905874	9897.12905874	-0.00000000000363797881
9.3000	10938.01920817	10938.01920817	0.00000000000000000000
9.4000	12088.38073022	12088.38073022	-0.00000000000363797881
9.5000	13359.72682966	13359.72682966	0.00000000000181898940
9.6000	14764.78156558	14764.78156558	-0.00000000000363797881
9.7000	16317.60719802	16317.60719802	-0.00000000000181898940
9.8000	18033.74492783	18033.74492783	-0.00000000000363797881
9.9000	19930.37043823	19930.37043823	-0.00000000000363797881
10.0000	22026.46579481	22026.46579481	-0.00000000000727595761

Here is a graph of the differences.



Analysis:
As seen by the graph and from my output, the differences appear to be larger as the value x distances itself from the starting point of 1. In other words, as x approaches the upper boundary set to 10, the differences between my function and math.h function of exp(x) increases. Again, this is very similar to the cases of sin(x) and cos(x) in which the more terms that are added to

the approximation, the more accurate the returned approximation will be. Like the lab document said, we can see that the approximation began to diverge significantly around x=7, showing that 10 terms are insufficient for an accurate approximation and more terms are needed.

OUTPUT for Log(x)

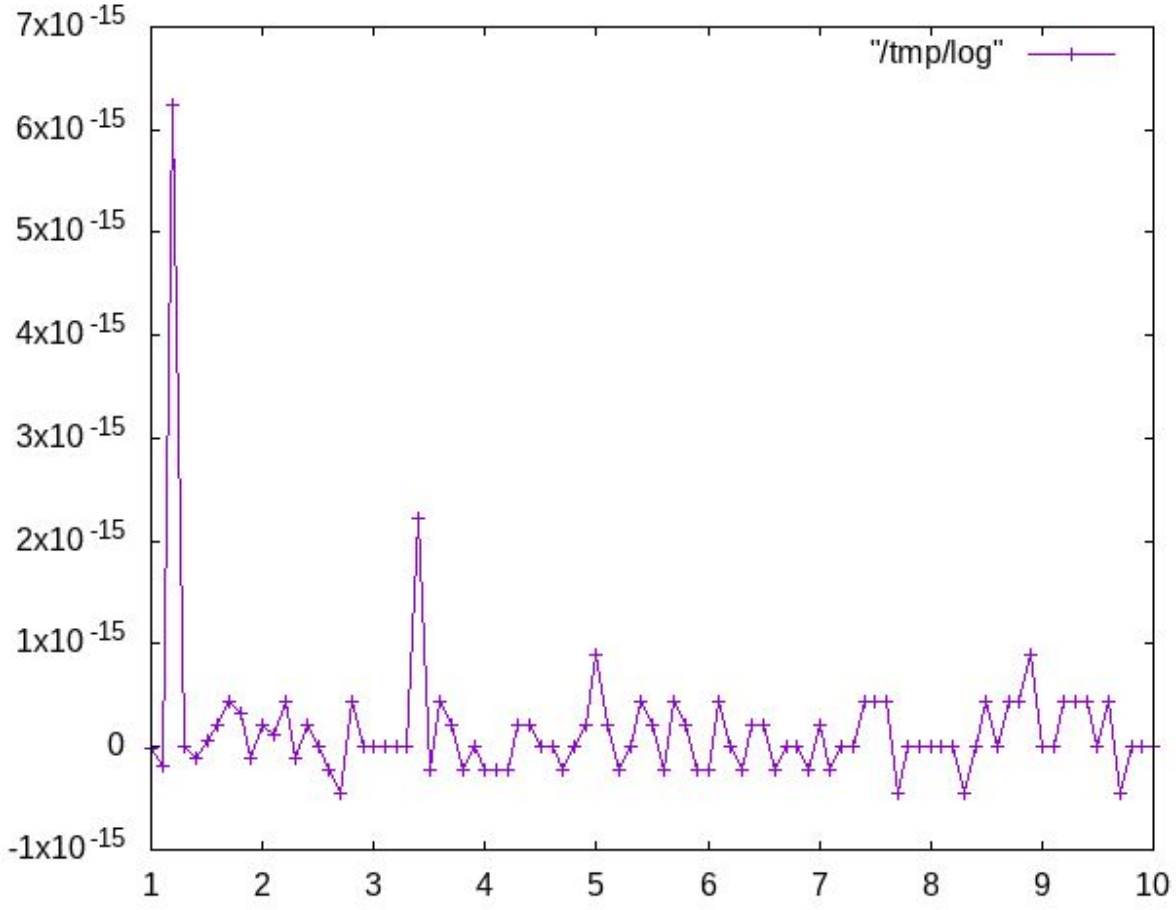
The output for running Log(x) in the specified range [1, 10) as shown in terminal is as follows.

x	Log	Library	Difference
1.0000	-0.00000000	0.00000000	-0.00000000000000000781
1.1000	0.09531018	0.09531018	-0.00000000000000018041
1.2000	0.18232156	0.18232156	0.00000000000000624500
1.3000	0.26236426	0.26236426	0.00000000000000000000
1.4000	0.33647224	0.33647224	-0.00000000000000011102
1.5000	0.40546511	0.40546511	0.00000000000000005551
1.6000	0.47000363	0.47000363	0.00000000000000022204
1.7000	0.53062825	0.53062825	0.00000000000000044409
1.8000	0.58778666	0.58778666	0.00000000000000033307
1.9000	0.64185389	0.64185389	-0.00000000000000011102
2.0000	0.69314718	0.69314718	0.00000000000000022204
2.1000	0.74193734	0.74193734	0.00000000000000011102
2.2000	0.78845736	0.78845736	0.00000000000000044409
2.3000	0.83290912	0.83290912	-0.00000000000000011102
2.4000	0.87546874	0.87546874	0.00000000000000022204
2.5000	0.91629073	0.91629073	0.00000000000000000000
2.6000	0.95551145	0.95551145	-0.00000000000000022204
2.7000	0.99325177	0.99325177	-0.00000000000000044409
2.8000	1.02961942	1.02961942	0.00000000000000044409
2.9000	1.06471074	1.06471074	0.00000000000000000000
3.0000	1.09861229	1.09861229	0.00000000000000000000
3.1000	1.13140211	1.13140211	0.00000000000000000000
3.2000	1.16315081	1.16315081	0.00000000000000000000
3.3000	1.19392247	1.19392247	0.00000000000000000000
3.4000	1.22377543	1.22377543	0.00000000000000222045
3.5000	1.25276297	1.25276297	-0.00000000000000022204
3.6000	1.28093385	1.28093385	0.00000000000000044409
3.7000	1.30833282	1.30833282	0.00000000000000022204
3.8000	1.33500107	1.33500107	-0.00000000000000022204
3.9000	1.36097655	1.36097655	0.00000000000000000000
4.0000	1.38629436	1.38629436	-0.00000000000000022204
4.1000	1.41098697	1.41098697	-0.00000000000000022204
4.2000	1.43508453	1.43508453	-0.00000000000000022204
4.3000	1.45861502	1.45861502	0.00000000000000022204
4.4000	1.48160454	1.48160454	0.00000000000000022204
4.5000	1.50407740	1.50407740	0.00000000000000000000

4.6000	1.52605630	1.52605630	0.00000000000000000000
4.7000	1.54756251	1.54756251	-0.00000000000000022204
4.8000	1.56861592	1.56861592	0.00000000000000000000
4.9000	1.58923521	1.58923521	0.00000000000000022204
5.0000	1.60943791	1.60943791	0.00000000000000088818
5.1000	1.62924054	1.62924054	0.00000000000000022204
5.2000	1.64865863	1.64865863	-0.00000000000000022204
5.3000	1.66770682	1.66770682	0.00000000000000000000
5.4000	1.68639895	1.68639895	0.00000000000000044409
5.5000	1.70474809	1.70474809	0.00000000000000022204
5.6000	1.72276660	1.72276660	-0.00000000000000022204
5.7000	1.74046617	1.74046617	0.00000000000000044409
5.8000	1.75785792	1.75785792	0.00000000000000022204
5.9000	1.77495235	1.77495235	-0.00000000000000022204
6.0000	1.79175947	1.79175947	-0.00000000000000022204
6.1000	1.80828877	1.80828877	0.00000000000000044409
6.2000	1.82454929	1.82454929	0.00000000000000000000
6.3000	1.84054963	1.84054963	-0.00000000000000022204
6.4000	1.85629799	1.85629799	0.00000000000000022204
6.5000	1.87180218	1.87180218	0.00000000000000022204
6.6000	1.88706965	1.88706965	-0.00000000000000022204
6.7000	1.90210753	1.90210753	0.00000000000000000000
6.8000	1.91692261	1.91692261	0.00000000000000000000
6.9000	1.93152141	1.93152141	-0.00000000000000022204
7.0000	1.94591015	1.94591015	0.00000000000000022204
7.1000	1.96009478	1.96009478	-0.00000000000000022204
7.2000	1.97408103	1.97408103	0.00000000000000000000
7.3000	1.98787435	1.98787435	0.00000000000000000000
7.4000	2.00148000	2.00148000	0.00000000000000044409
7.5000	2.01490302	2.01490302	0.00000000000000044409
7.6000	2.02814825	2.02814825	0.00000000000000044409
7.7000	2.04122033	2.04122033	-0.00000000000000044409
7.8000	2.05412373	2.05412373	0.00000000000000000000
7.9000	2.06686276	2.06686276	0.00000000000000000000
8.0000	2.07944154	2.07944154	0.00000000000000000000
8.1000	2.09186406	2.09186406	0.00000000000000000000
8.2000	2.10413415	2.10413415	0.00000000000000000000
8.3000	2.11625551	2.11625551	-0.00000000000000044409
8.4000	2.12823171	2.12823171	0.00000000000000000000
8.5000	2.14006616	2.14006616	0.00000000000000044409
8.6000	2.15176220	2.15176220	0.00000000000000000000
8.7000	2.16332303	2.16332303	0.00000000000000044409
8.8000	2.17475172	2.17475172	0.00000000000000044409
8.9000	2.18605128	2.18605128	0.00000000000000088818
9.0000	2.19722458	2.19722458	0.00000000000000000000

9.1000	2.20827441	2.20827441	0.00000000000000000000
9.2000	2.21920348	2.21920348	0.00000000000000044409
9.3000	2.23001440	2.23001440	0.00000000000000044409
9.4000	2.24070969	2.24070969	0.00000000000000044409
9.5000	2.25129180	2.25129180	0.00000000000000000000
9.6000	2.26176310	2.26176310	0.00000000000000044409
9.7000	2.27212589	2.27212589	-0.00000000000000044409
9.8000	2.28238239	2.28238239	0.00000000000000000000
9.9000	2.29253476	2.29253476	0.00000000000000000000
10.0000	2.30258509	2.30258509	0.00000000000000000000

Here is a graph of the differences.



Analysis:

As seen by the graph and from my output, the differences appear to be larger as the value x approaches 1 which is the complete opposite of exp(x). In other words, as x approaches the lower boundary set to 1, the differences between my function and math.h function of log(x) increases. When implementing this method, we saw that each iteration of Newton's method produced better and more accurate approximations. Therefore, the differences between these two functions could have been that my implementation simply did not iterate enough times to yield a more accurate approximation as the value of x approaches 1. In addition, note that our

implementation of Log(x) also utilized Taylor Expansion method through the use of Exp(x). This could have also contributed to the various differences found.

Conclusion:

After analyzing the data, I found that while implementing the Taylor Expansion method and Newton's method can yield accurate approximations, it is less accurate as x approaches certain bounds. For the Taylor Expansion method, we were only able to calculate a finite number of terms to yield as accurate of an approximation that we could compute. When approaching the boundaries, there was a need for an even greater number of terms that needed to be calculated in order to yield a more accurate approximation. Taylor Expansion method after all was meant to be used to calculate an infinite number of terms, but in our program, that was not possible, and we had to be satisfied with calculating a finite number of terms. Newton's method was also the same in the sense that the greater the number of iterations, the more accurate the representation would be. However, because our function also utilized Taylor Expansion method through the Exp(x) function that we created, there was clearly room for error when computing the final approximation as the value of x approaches 1. This could have been a possible reason behind the differences shown in the graph.