University of Puerto Rico Rio Piedras

Department of Physics

Bessel Functions

Application

Methods of Theoretical Physics

Luis A. Flores Carrubio

841-10-2500

12/4/2019

A variety of special functions are encountered in the solution of physical problems, particularly in the solution of differential equations. These include Legendre and Hermite polynomials, Bessel functions, spherical harmonics, and others. A special type of function known as Bessel functions are seen in problems with electric fields, vibrations, heat conduction, optical diffraction, and others. These functions were first defined by German mathematicians D. Bernoulli and F. Bessel and are obtained from the Bessel differential equation.

$$x^2 \frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - v^2)y = 0 \tag{1}$$

The solution to the Bessel equation is found by Frobenius method and the solution is known as the Bessel function of the first kind, normally labeled J_n. The functions $J_n(x)$ are those that for non-negative integer n (order of Bessel function), are regular at x=0. Bessel functions are then defined by the series expansion around x=0 where the gamma function $\Gamma(x) = (x-1)!$ is used:

$$J_n(x) = x^n \sum_{m=0}^{\infty} \frac{-1^m x^{2m}}{2^{2m+n} m!\, \Gamma(n+m+1)}\,. \tag{2}$$

Bessel functions when graphed (*fig1)* have a similarity to sine and cosine functions as they oscillate but are not necessarily periodic. Furthermore, they satisfy recurrence relations which can be found from the generating function *(eq3)*. The recurrence relation for Bessel functions of the first kind is given by equation *(eq4) [Krey]*.

$$g(x,t) = e^{\frac{x}{2}(t-\frac{1}{t})} = \sum_{-\infty}^{\infty} J_n(x)\, t^n \tag{3}$$

$$J_{n-1}(x) + J_{n+1}(x) = \frac{2n}{x} J_n(x) \tag{4}$$

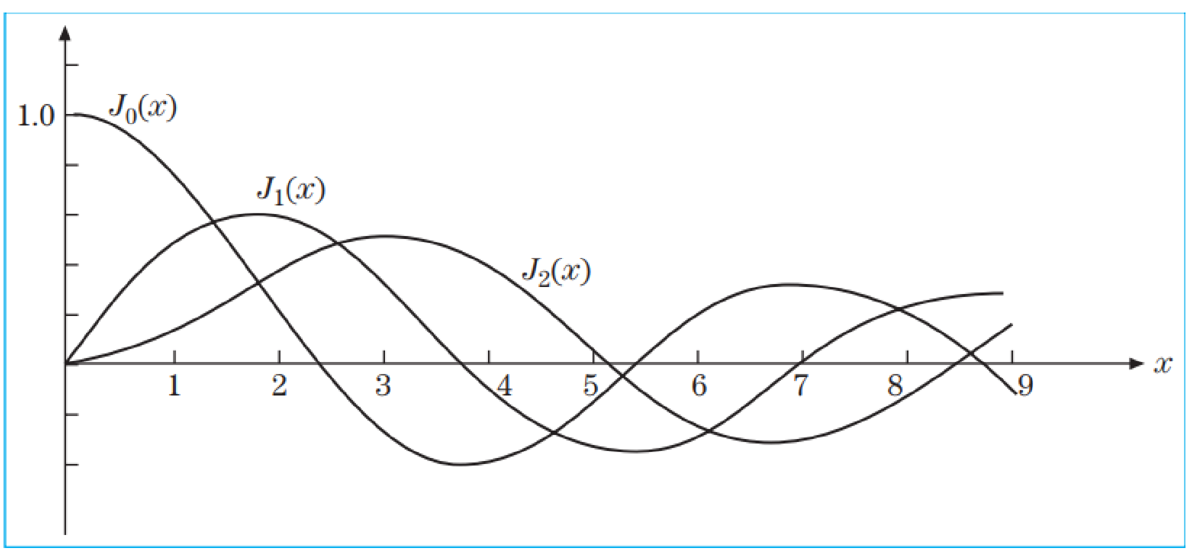

Figure 1. Graphs of the Bessel functions $J_0, J_1, and\ J_2$. [Krey]

A computer application was created with the purpose of calculating the Bessel functions for various arguments of x and order of polynomials. Therefore, the tables *(table1), (table2),* and *(table3)* were included as reference to verify if the application can find the appropriate values for the functions given the user's parameters. The application was made using the programming language *Visual Basic* and several precautions were taken to avoid problems with arithmetic overflowing. For example, the *decimal* data type was used instead of *double* to avoid rounding errors in the approximations of the Bessel functions. However, the tradeoff is that the *decimal* data type holds smaller numbers than the *double* and might result in an overflow depending on the parameters chosen by the user which are also validated by the program.

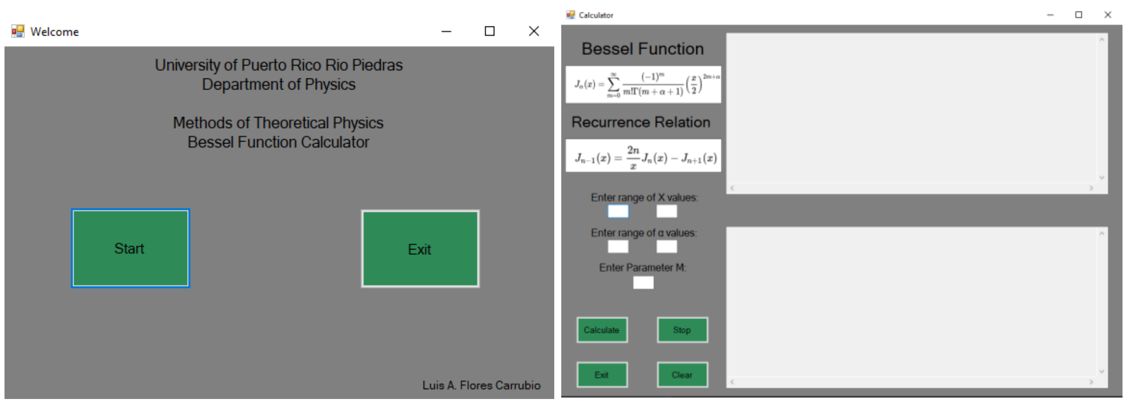

Figure 2. Welcome and calculator windows of the Bessel Function Application.

The application starts with a welcome screen and goes into the calculator window when the user presses the start button *(fig2)*. In the calculator window the user is asked to enter the range for the values of the argument x and the order n of the polynomials. It also asks the user to enter the maximum value for the m summation index used in equation *(eq2)*. As a precaution, these values will be validated by the program to avoid overflow issues. However, the application is more than capable of calculating the Bessel functions from $x=0$ to $x=1$ for orders of $n=0$ to $n=10$ with $m=10$. Once the user enters the necessary values and presses the calculate button, the program will display the values of the Bessel functions in the windows to the right. The calculations can be stopped at any time by pressing the stop button. The upper window displays the values from using equation *(eq2)* while the lower window displays approximations using the recurrence formula *(eq4)*. The recurrence formula was adjusted, and the Bessel functions calculated as

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x)\,. \tag{5}$$

The results of Bessel functions obtained from equations *(eq2)* and *(eq5)* can be seen in figure *(fig3)* for polynomials up to the fourth order while *(fig4)* displays the polynomials up to the tenth order. The range of the x values goes from zero to one with a step of 0.1 therefore, the Bessel functions for these values are displayed. In the recurrence formula *(eq5)* division by zero is not possible, therefore the values for *x=0* in the lower result display were omitted.

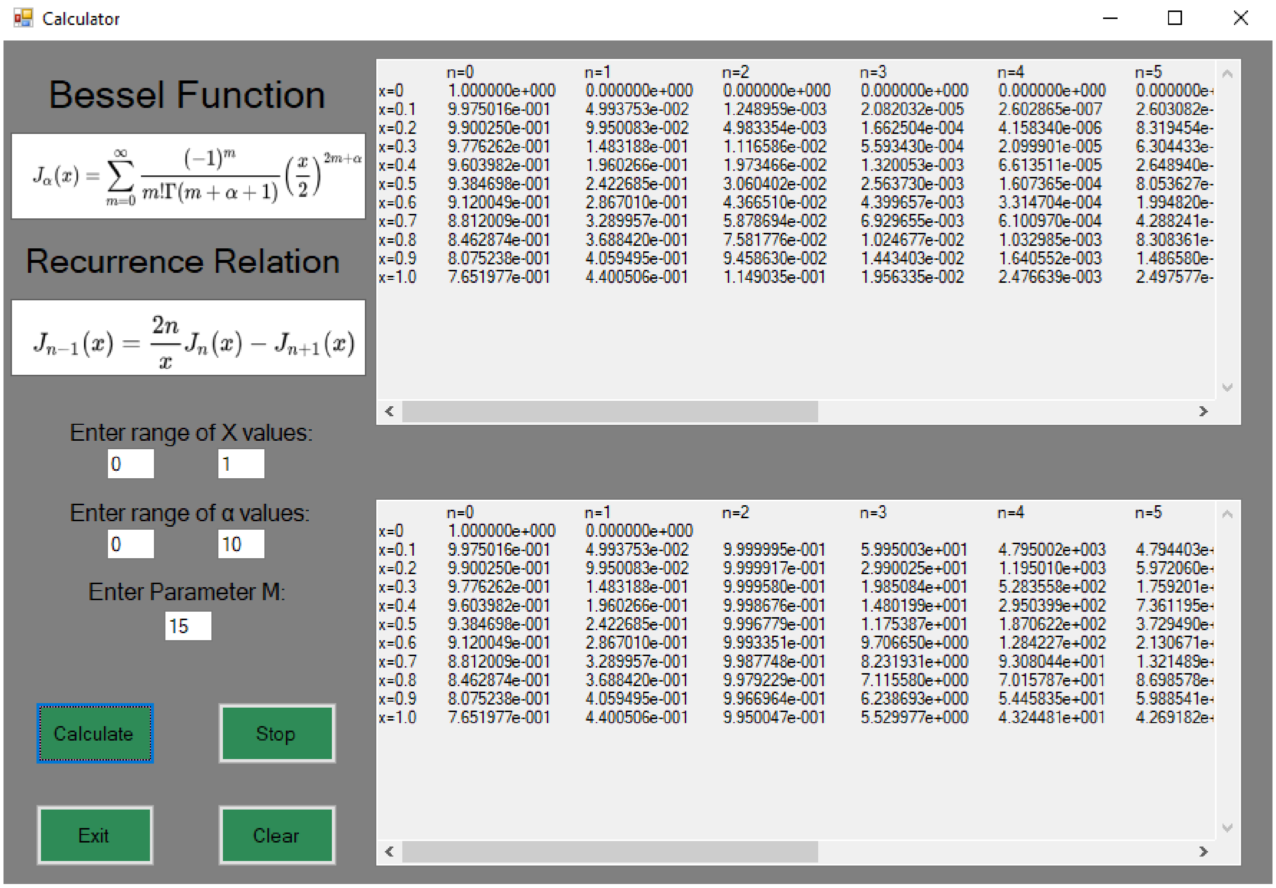

Figure 3. Bessel functions up to the fourth polynomial using *(eq2)* and *(eq5)*.

For higher order polynomials the results can be seen in *(fig4)* for both equations *(eq2)* and *(eq5)*. The application can calculate higher order polynomials; however, it seems enough to present the results until the tenth order polynomial. Furthermore, it can calculate higher arguments than *x=1 (fig5).* From inspecting and comparing the results obtained from equations *(eq2)* and *(eq5)* the values differ incredibly. Given that $J_0, and\ J_1$ must be known for the recurrence formula, the values start to differ once the polynomial of second order is calculated. Comparing these values to those of *(table1), (table2), and (table3)* it is evident that *(eq2)* works while the recurrence relation *(eq5)* fails to provide a good approximation of the Bessel functions. While it is possible to calculate a percentage of error for *(eq5),* we can see that the exponents are different compared to those of *(eq2)* hence making it unnecessary and evident that the errors are great.

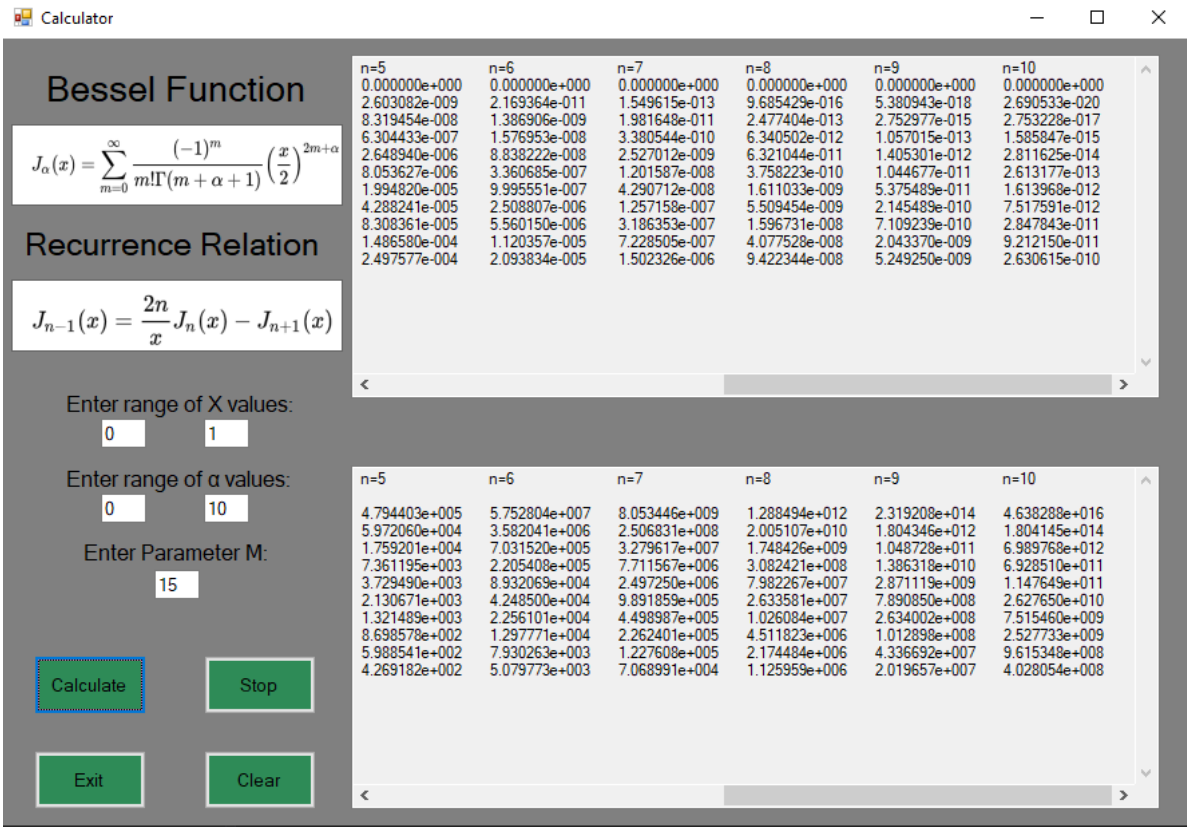

Figure 4. Bessel functions up to the tenth polynomial using *(eq2)* and *(eq5)*.

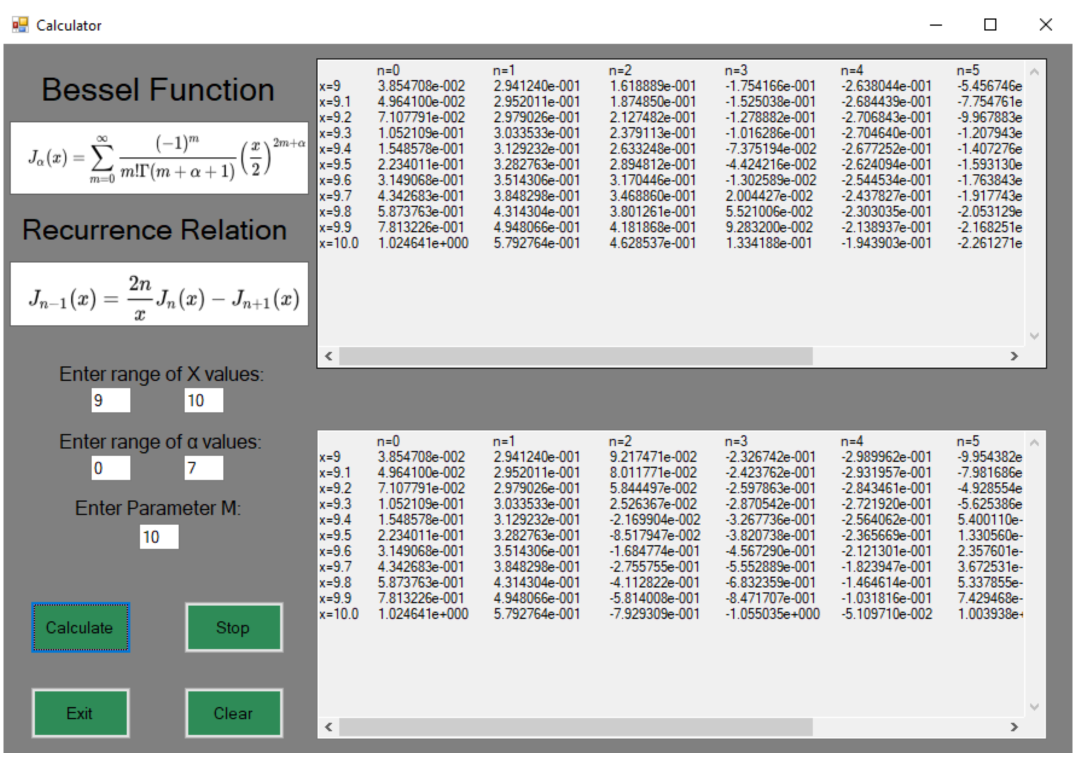

Figure 5. Bessel functions up to the 7th polynomial using *(eq2)* and *(eq5) with x=9 to 10*.

A possible solution to obtain less error from the recurrence formula is by using Miller's recurrence algorithm which is a procedure for calculating a rapidly decreasing solution of a linear recurrence relation. This algorithm applies to Bessel functions of the first kind and other special functions. However, instead of using the recurrence formula *(eq5)* in the forward direction as the program does, Miller's algorithm uses the backwards direction decreasing the rounding error propagation *(eq6)*.

$$J_{n-1}(x) = \frac{2n}{x} J_n(x) - J_{n+1}(x) \tag{6}$$

By using Miller's algorithm, good approximations of the Bessel functions are guaranteed, and comparisons can be made with the values in the program obtained from *(eq2)*. All tries to approximate Bessel functions using a forward recurrence formula introduce errors [Hart]. As an example, the value for $J_6(1) = 2.093834e - 005$ using *(eq2)* but by using *(eq5)* we obtain $J_6(1) = 5.079773e + 003$. Therefore, for future work Miller's algorithm should be coded into the program and verify if the approximations are reliable.

	n=0	n=1	n=2	n=3	n=4
x=0	1.000000e+000	0.000000e+000	0.000000e+000	0.000000e+000	0.000000e+000
x=0.1	9.975016e-001	4.993753e-002	1.248959e-003	2.082032e-005	2.602865e-007
x=0.2	9.900250e-001	9.950083e-002	4.983354e-003	1.662504e-004	4.158340e-006
x=0.3	9.776262e-001	1.483188e-001	1.116586e-002	5.593430e-004	2.099901e-005
x=0.4	9.603982e-001	1.960266e-001	1.973466e-002	1.320053e-003	6.613511e-005
x=0.5	9.384698e-001	2.422685e-001	3.060402e-002	2.563730e-003	1.607365e-004
x=0.6	9.120049e-001	2.867010e-001	4.366510e-002	4.399657e-003	3.314704e-004
x=0.7	8.812009e-001	3.289957e-001	5.878694e-002	6.929655e-003	6.100970e-004
x=0.8	8.462874e-001	3.688420e-001	7.581776e-002	1.024677e-002	1.032985e-003
x=0.9	8.075238e-001	4.059495e-001	9.458630e-002	1.443403e-002	1.640552e-003
x=1.0	7.651977e-001	4.400506e-001	1.149035e-001	1.956335e-002	2.476639e-003

n=5	n=6	n=7	n=8	n=9	n=10
0.000000e+000	0.000000e+000	0.000000e+000	0.000000e+000	0.000000e+000	0.000000e+000
2.603082e-009	2.169364e-011	1.549615e-013	9.685429e-016	5.380943e-018	2.690533e-020
8.319454e-008	1.386906e-009	1.981648e-011	2.477404e-013	2.752977e-015	2.753228e-017
6.304433e-007	1.576953e-008	3.380544e-010	6.340502e-012	1.057015e-013	1.585847e-015
2.648940e-006	8.838222e-008	2.527012e-009	6.321044e-011	1.405301e-012	2.811625e-014
8.053627e-006	3.360685e-007	1.201587e-008	3.758223e-010	1.044677e-011	2.613177e-013
1.994820e-005	9.995551e-007	4.290712e-008	1.611033e-009	5.375489e-011	1.613968e-012
4.288241e-005	2.508807e-006	1.257158e-007	5.509454e-009	2.145489e-010	7.517591e-012
8.308361e-005	5.560150e-006	3.186353e-007	1.596731e-008	7.109239e-010	2.847843e-011
1.486580e-004	1.120357e-005	7.228505e-007	4.077528e-008	2.043370e-009	9.212150e-011
2.497577e-004	2.093834e-005	1.502326e-006	9.422344e-008	5.249250e-009	2.630615e-010

Table 4. Table displaying the values of the program for (eq2)

	n=0	n=1	n=2	n=3	n=4
x=0	1.000000e+000	0.000000e+000			
x=0.1	9.975016e-001	4.993753e-002	9.999995e-001	5.995003e+001	4.795002e+003
x=0.2	9.900250e-001	9.950083e-002	9.999917e-001	2.990025e+001	1.195010e+003
x=0.3	9.776262e-001	1.483188e-001	9.999580e-001	1.985084e+001	5.283558e+002
x=0.4	9.603982e-001	1.960266e-001	9.998676e-001	1.480199e+001	2.950399e+002
x=0.5	9.384698e-001	2.422685e-001	9.996779e-001	1.175387e+001	1.870622e+002
x=0.6	9.120049e-001	2.867010e-001	9.993351e-001	9.706650e+000	1.284227e+002
x=0.7	8.812009e-001	3.289957e-001	9.987748e-001	8.231931e+000	9.308044e+001
x=0.8	8.462874e-001	3.688420e-001	9.979229e-001	7.115580e+000	7.015787e+001
x=0.9	8.075238e-001	4.059495e-001	9.966964e-001	6.238693e+000	5.445835e+001
x=1.0	7.651977e-001	4.400506e-001	9.950047e-001	5.529977e+000	4.324481e+001

n=5	n=6	n=7	n=8	n=9	n=10
4.794403e+005	5.752804e+007	8.053446e+009	1.288494e+012	2.319208e+014	4.638288e+016
5.972060e+004	3.582041e+006	2.506831e+008	2.005107e+010	1.804346e+012	1.804145e+014
1.759201e+004	7.031520e+005	3.279617e+007	1.748426e+009	1.048728e+011	6.989768e+012
7.361195e+003	2.205408e+005	7.711567e+006	3.082421e+008	1.386318e+010	6.928510e+011
3.729490e+003	8.932069e+004	2.497250e+006	7.982267e+007	2.871119e+009	1.147649e+011
2.130671e+003	4.248500e+004	9.891859e+005	2.633581e+007	7.890850e+008	2.627650e+010
1.321489e+003	2.256101e+004	4.498987e+005	1.026084e+007	2.634002e+008	7.515460e+009
8.698578e+002	1.297771e+004	2.262401e+005	4.511823e+006	1.012898e+008	2.527733e+009
5.988541e+002	7.930263e+003	1.227608e+005	2.174484e+006	4.336692e+007	9.615348e+008
4.269182e+002	5.079773e+003	7.068991e+004	1.125959e+006	2.019657e+007	4.028054e+008

Table 4. Table displaying the values of the program for the recurrence equation (eq5).

References

[Krey] Kreyzig E. (1999). *Advanced Engineering Mathematics 8th*. Wiley.

[Math] Abramowitz M. & Stegun I. (1970). *Handbook of Mathematical Functions* 9th. Retrieved http://www.plouffe.fr/simon/Dictionnaires/Handbook%20of%20mathematical%20functions%20by%20M.%20Abramowitz%20&%20I.%20Stegun.pdf

[Hart] Hart, J.F. (1978). *Computer Approximations* (reprint ed.). Malabar, Florida: Robert E. Krieger. pp. 25–26. ISBN 978-0-88275-642-4.

Appendix

x	$J_0(x)$	$J_1(x)$	$J_2(x)$
0.0	1.00000 00000 00000	0.00000 00000	0.00000 00000
0.1	0.99750 15620 66040	0.04993 75260	0.00124 89587
0.2	0.99002 49722 39576	0.09950 08326	0.00498 33542
0.3	0.97762 62465 38296	0.14831 88163	0.01116 58619
0.4	0.96039 82266 59563	0.19602 65780	0.01973 46631
0.5	0.93846 98072 40813	0.24226 84577	0.03060 40235
0.6	0.91200 48634 97211	0.28670 09881	0.04366 50967
0.7	0.88120 08886 07405	0.32899 57415	0.05878 69444
0.8	0.84628 73527 50480	0.36884 20461	0.07581 77625
0.9	0.80752 37981 22545	0.40594 95461	0.09458 63043
1.0	0.76519 76865 57967	0.44005 05857	0.11490 34849
1.1	0.71962 20185 27511	0.47090 23949	0.13656 41540
1.2	0.67113 27442 64363	0.49828 90576	0.15934 90183
1.3	0.62008 59895 61509	0.52202 32474	0.18302 66988
1.4	0.56685 51203 74289	0.54194 77139	0.20735 58995
1.5	0.51182 76717 35918	0.55793 65079	0.23208 76721
1.6	0.45540 21676 39381	0.56989 59353	0.25696 77514
1.7	0.39798 48594 46109	0.57776 52315	0.28173 89424
1.8	0.33998 64110 42558	0.58151 69517	0.30614 35353
1.9	0.28181 85593 74385	0.58115 70727	0.32992 57277
2.0	0.22389 07791 41236	0.57672 48078	0.35283 40286
2.1	0.16660 69803 31990	0.56829 21358	0.37462 36252
2.2	0.11036 22669 22174	0.55596 30498	0.39505 86875
2.3	0.05553 97844 45602	0.53987 25326	0.41391 45917
2.4	+0.00250 76832 97244	0.52018 52682	0.43098 00402
2.5	-0.04838 37764 68198	0.49709 41025	0.44605 90584
2.6	-0.09680 49543 97038	0.47081 82665	0.45897 28517
2.7	-0.14244 93700 46012	0.44160 13791	0.46956 15027
2.8	-0.18503 60333 64387	0.40970 92469	0.47768 54954
2.9	-0.22431 15457 91968	0.37542 74818	0.48322 70505
3.0	-0.26005 19549 01933	0.33905 89585	0.48609 12606
3.1	-0.29206 43476 50698	0.30092 11331	0.48620 70142
3.2	-0.32018 81696 57123	0.26134 32488	0.48352 77001
3.3	-0.34429 62603 98885	0.22066 34530	0.47803 16865
3.4	-0.36429 55967 62000	0.17922 58517	0.46972 25683
3.5	-0.38012 77399 87263	0.13737 75274	0.45862 91842
3.6	-0.39176 89837 00798	0.09546 55472	0.44480 53988
3.7	-0.39923 02033 71191	0.05383 39877	0.42832 96562
3.8	-0.40255 64101 78564	+0.01282 10029	0.40930 43065
3.9	-0.40182 60148 87640	-0.02724 40396	0.38785 47125
4.0	-0.39714 98098 63847	-0.06604 33280	0.36412 81459
4.1	-0.38866 96798 35854	-0.10327 32577	0.33829 24809
4.2	-0.37655 70543 67568	-0.13864 69421	0.31053 47010
4.3	-0.36101 11172 36535	-0.17189 65602	0.28105 92288
4.4	-0.34225 67900 03886	-0.20277 55219	0.25008 60982
4.5	-0.32054 25089 85121	-0.23106 04319	0.21784 89837
4.6	-0.29613 78165 74141	-0.25655 28361	0.18459 31052
4.7	-0.26933 07894 19753	-0.27908 07358	0.15057 30295
4.8	-0.24042 53272 91183	-0.29849 98581	0.11605 03864
4.9	-0.20973 83275 85326	-0.31469 46710	0.08129 15231
5.0	-0.17759 67713 14338	-0.32757 91376	0.04656 51163
	$\left[\begin{smallmatrix}(-4)6\\11\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}(-4)5\\8\end{smallmatrix}\right]$	$\left[\begin{smallmatrix}(-4)3\\7\end{smallmatrix}\right]$

$$J_{n+1}(x)=\frac{2n}{x}J_n(x)-J_{n-1}(x)$$

Compiled from British Association for the Advancement of Science, Bessel functions, Part II. Functions of positive integer order, Mathematical Tables, vol. X (Cambridge Univ. Press, Cambridge, England, 1952) and Harvard Computation Laboratory, Tables of the Bessel functions of the first kind of orders 0 through 135, vols. 3–14 (Harvard Univ. Press, Cambridge, Mass., 1947-1951) (with permission).

Table 1. Values of the Bessel functions for orders $J_0, J_1, and\ J_2$ and various argument values for x. [Math]

x	$J_3(x)$	$J_4(x)$	$J_5(x)$	$J_6(x)$	$J_7(x)$	$J_8(x)$	$J_9(x)$
0.0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.2	(-4)1.6625	(-6)4.1583	(-8)8.3195	(-9)1.3869	(-11)1.9816	(-13)2.4774	(-15)2.7530
0.4	(-3)1.3201	(-5)6.6135	(-6)2.6489	(-8)8.8382	(- 9)2.5270	(-11)6.3210	(-12)1.4053
0.6	(-3)4.3997	(-4)3.3147	(-5)1.9948	(-7)9.9956	(- 8)4.2907	(- 9)1.6110	(-11)5.3755
0.8	(-2)1.0247	(-3)1.0330	(-5)8.3084	(-6)5.5601	(- 7)3.1864	(- 8)1.5967	(-10)7.1092
1.0	(-2)1.9563	(-3)2.4766	(-4)2.4976	(-5)2.0938	(- 6)1.5023	(- 8)9.4223	(- 9)5.2493
1.2	(-2)3.2874	(-3)5.0227	(-4)6.1010	(-5)6.1541	(- 6)5.3093	(- 7)4.0021	(- 8)2.6788
1.4	(-2)5.0498	(-3)9.0629	(-3)1.2901	(-4)1.5231	(- 5)1.5366	(- 6)1.3538	(- 7)1.0587
1.6	(-2)7.2523	(-2)1.4995	(-3)2.4524	(-4)3.3210	(- 5)3.8397	(- 6)3.8744	(- 7)3.4687
1.8	(-2)9.8802	(-2)2.3197	(-3)4.2936	(-4)6.5690	(- 5)8.5712	(- 6)9.7534	(- 7)9.8426
2.0	0.12894	(-2)3.3996	(-3)7.0396	(-3)1.2024	(- 4)1.7494	(- 5)2.2180	(- 6)2.4923
2.2	0.16233	(-2)4.7647	(-2)1.0937	(-3)2.0660	(- 4)3.3195	(- 5)4.6434	(- 6)5.7535
2.4	0.19811	(-2)6.4307	(-2)1.6242	(-3)3.3669	(- 4)5.9274	(- 5)9.0756	(- 5)1.2300
2.6	0.23529	(-2)8.4013	(-2)2.3207	(-3)5.2461	(- 3)1.0054	(- 4)1.6738	(- 5)2.4647
2.8	0.27270	(-1)1.0667	(-2)3.2069	(-3)7.8634	(- 3)1.6314	(- 4)2.9367	(- 5)4.6719
3.0	0.30906	0.13203	(-2)4.3028	(-2)1.1394	(- 3)2.5473	(- 4)4.9344	(- 5)8.4395
3.2	0.34307	0.15972	(-2)5.6238	(-2)1.6022	(- 3)3.8446	(- 4)7.9815	(- 4)1.4615
3.4	0.37339	0.18920	(-2)7.1785	(-2)2.1934	(- 3)5.6301	(- 3)1.2482	(- 4)2.4382
3.6	0.39876	0.21980	(-2)8.9680	(-2)2.9311	(- 3)8.0242	(- 3)1.8940	(- 4)3.9339
3.8	0.41803	0.25074	(-1)1.0984	(-2)3.8316	(- 2)1.1159	(- 3)2.7966	(- 4)6.1597
4.0	0.43017	0.28113	0.13209	(-2)4.9088	(- 2)1.5176	(- 3)4.0287	(- 4)9.3860
4.2	0.43439	0.31003	0.15614	(-2)6.1725	(- 2)2.0220	(- 3)5.6739	(- 3)1.3952
4.4	0.43013	0.33645	0.18160	(-2)7.6279	(- 2)2.6433	(- 3)7.8267	(- 3)2.0275
4.6	0.41707	0.35941	0.20799	(-2)9.2745	(- 2)3.3953	(- 2)1.0591	(- 3)2.8852
4.8	0.39521	0.37796	0.23473	(-1)1.1105	(- 2)4.2901	(- 2)1.4079	(- 3)4.0270
5.0	0.36483	0.39123	0.26114	0.13105	(- 2)5.3376	(- 2)1.8405	(- 3)5.5203
5.2	0.32652	0.39847	0.28651	0.15252	(- 2)6.5447	(- 2)2.3689	(- 3)7.4411
5.4	0.28113	0.39906	0.31007	0.17515	(- 2)7.9145	(- 2)3.0044	(- 3)9.8734
5.6	0.22978	0.39257	0.33103	0.19856	(- 2)9.4455	(- 2)3.7577	(- 2)1.2907
5.8	0.17382	0.37877	0.34862	0.22230	(- 1)1.1131	(- 2)4.6381	(- 2)1.6639
6.0	0.11477	0.35764	0.36209	0.24584	0.12959	(- 2)5.6532	(- 2)2.1165
6.2	+0.05428	0.32941	0.37077	0.26860	0.14910	(- 2)6.8077	(- 2)2.6585
6.4	-0.00591	0.29453	0.37408	0.28996	0.16960	(- 2)8.1035	(- 2)3.2990
6.6	-0.06406	0.25368	0.37155	0.30928	0.19077	(- 2)9.5385	(- 2)4.0468
6.8	-0.11847	0.20774	0.36288	0.32590	0.21224	(- 1)1.1107	(- 2)4.9093
7.0	-0.16756	0.15780	0.34790	0.33920	0.23358	0.12797	(- 2)5.8921
7.2	-0.20987	0.10509	0.32663	0.34857	0.25432	0.14594	(- 2)6.9987
7.4	-0.24420	+0.05097	0.29930	0.35349	0.27393	0.16476	(- 2)8.2300
7.6	-0.26958	-0.00313	0.26629	0.35351	0.29188	0.18417	(- 2)9.5839
7.8	-0.28535	-0.05572	0.22820	0.34828	0.30762	0.20385	(- 1)1.1054
8.0	-0.29113	-0.10536	0.18577	0.33758	0.32059	0.22345	0.12632
8.2	-0.28692	-0.15065	0.13994	0.32131	0.33027	0.24257	0.14303
8.4	-0.27302	-0.19033	0.09175	0.29956	0.33619	0.26075	0.16049
8.6	-0.25005	-0.22326	+0.04237	0.27253	0.33790	0.27755	0.17847
8.8	-0.21896	-0.24854	-0.00699	0.24060	0.33508	0.29248	0.19670
9.0	-0.18094	-0.26547	-0.05504	0.20432	0.32746	0.30507	0.21488
9.2	-0.13740	-0.27362	-0.10053	0.16435	0.31490	0.31484	0.23266
9.4	-0.08997	-0.27284	-0.14224	0.12152	0.29737	0.32138	0.24965
9.6	-0.04034	-0.26326	-0.17904	0.07676	0.27499	0.32427	0.26546
9.8	+0.00970	-0.24528	-0.20993	+0.03107	0.24797	0.32318	0.27967
10.0	0.05838	-0.21960	-0.23406	-0.01446	0.21671	0.31785	0.29186

Compiled from British Association for the Advancement of Science, Bessel functions, Part II. Functions of positive integer order, Mathematical Tables, vol. X (Cambridge Univ. Press, Cambridge, England, 1952) and Mathematical Tables Project, Table of $f_n(x)=n!(\frac{1}{2}x)^{-n}J_n(x)$. J. Math. Phys. **23,** 45–60 (1944) (with permission).

Table 2. Values of the Bessel functions for higher orders and various argument values for x. [Math]

n	$J_n(1)$	$J_n(2)$	$J_n(5)$
0	(− 1)7.65197 6866	(− 1)2.23890 7791	(− 1)−1.77596 7713
1	(− 1)4.40050 5857	(− 1)5.76724 8078	(− 1)−3.27579 1376
2	(− 1)1.14903 4849	(− 1)3.52834 0286	(− 2)+4.65651 1628
3	(− 2)1.95633 5398	(− 1)1.28943 2495	(− 1) 3.64831 2306
4	(− 3)2.47663 8964	(− 2)3.39957 1981	(− 1) 3.91232 3605
5	(− 4)2.49757 7302	(− 3)7.03962 9756	(− 1) 2.61140 5461
6	(− 5)2.09383 3800	(− 3)1.20242 8972	(− 1) 1.31048 7318
7	(− 6)1.50232 5817	(− 4)1.74944 0749	(− 2) 5.33764 1016
8	(− 8)9.42234 4173	(− 5)2.21795 5229	(− 2) 1.84052 1665
9	(− 9)5.24925 0180	(− 6)2.49234 3435	(− 3) 5.52028 3139
10	(− 10)2.63061 5124	(− 7)2.51538 6283	(− 3) 1.46780 2647
11	(− 11)1.19800 6746	(− 8)2.30428 4758	(− 4) 3.50927 4498
12	(− 13)4.99971 8179	(− 9)1.93269 5149	(− 5) 7.62781 3166
13	(− 14)1.92561 6764	(− 10)1.49494 2010	(− 5) 1.52075 8221
14	(− 16)6.88540 8200	(− 11)1.07294 6448	(− 6) 2.80129 5810
15	(− 17)2.29753 1532	(− 13)7.18301 6356	(− 7) 4.79674 3278
16	(− 19)7.18639 6587	(− 14)4.50600 5896	(− 8) 7.67501 5694
17	(− 20)2.11537 5568	(− 15)2.65930 7805	(− 8) 1.15266 7666
18	(− 22)5.88034 4574	(− 16)1.48173 7249	(− 9) 1.63124 4339
19	(− 23)1.54847 8441	(− 18)7.81924 3273	(− 10) 2.18282 5842
20	(− 25)3.87350 3009	(− 19)3.91897 2805	(− 11) 2.77033 0052
30	(− 42)3.48286 9794	(− 33)3.65025 6266	(− 21) 2.67117 7278
40	(− 60)1.10791 5851	(− 48)1.19607 7458	(− 33) 8.70224 1617
50	(− 80)2.90600 4948	(− 65)3.22409 5839	(− 45) 2.29424 7616
100	(−189)8.43182 8790	(−158)1.06095 3112	(−119) 6.26778 9396

n	$J_n(10)$	$J_n(50)$	$J_n(100)$
0	(− 1)−2.45935 7645	(− 2)+5.58123 2767	(−2)+1.99858 5030
1	(− 2)+4.34727 4617	(− 2)−9.75118 2813	(−2)−7.71453 5201
2	(− 1)+2.54630 3137	(− 2)−5.97128 0079	(−2)−2.15287 5734
3	(− 2)+5.83793 7931	(− 2)+9.27348 0406	(−2)+7.62842 0172
4	(− 1)−2.19602 6861	(− 2)+7.08409 7728	(−2)+2.61058 0945
5	(− 1)−2.34061 5282	(− 2)−8.14002 4770	(−2)−7.41957 3696
6	(− 2)−1.44588 4208	(− 2)−8.71210 2682	(−2)−3.35253 8314
7	(− 1)+2.16710 9177	(− 2)+6.04912 0126	(−2)+7.01726 9099
8	(− 1) 3.17854 1268	(− 1)+1.04058 5632	(−2)+4.33495 5988
9	(− 1) 2.91855 6853	(− 2)−2.71924 6104	(−2)−6.32367 6141
10	(− 1) 2.07486 1066	(− 1)−1.13847 8491	(−2)−5.47321 7694
11	(− 1) 1.23116 5280	(− 2)−1.83466 7862	(−2)+5.22903 2602
12	(− 2) 6.33702 5497	(− 1)+1.05775 3106	(−2)+6.62360 4866
13	(− 2) 2.89720 8393	(− 2)+6.91188 2768	(−2)−3.63936 7434
14	(− 2) 1.19571 6324	(− 2)−6.98335 2016	(−2)−7.56984 0399
15	(− 3) 4.50797 3144	(− 1)−1.08225 5990	(−2)+1.51981 2122
16	(− 3) 1.56675 6192	(− 3)+4.89816 0778	(−2)+8.02578 4036
17	(− 4) 5.05646 6697	(− 1)+1.11360 4219	(−2)+1.04843 8769
18	(− 4) 1.52442 4853	(− 2)+7.08269 2610	(−2)−7.66931 4854
19	(− 5) 4.31462 7752	(− 2)−6.03650 3508	(−2)−3.80939 2116
20	(− 5) 1.15133 6925	(− 1)−1.16704 3528	(−2)+6.22174 5850
30	(−12) 1.55109 6078	(− 2)+4.84342 5725	(−2)+8.14601 2958
40	(−21) 6.03089 5312	(− 1)−1.38176 2812	(−2)+7.27017 5482
50	(−30) 1.78451 3608	(− 1)+1.21409 0219	(−2)−3.86983 3973
100	(−89) 6.59731 6064	(−21)+1.11592 7368	(−2)+9.63666 7330

Table 3. Values of the Bessel functions for various orders and arguments of 1, 2, 5, 10, 50, and 100.

[Math]

Code

```vb
'Luis A. Flores Carrubio (841-10-2500)
'University of Puerto Rico Rio Piedras
'Physics Department FISI6401
'Bessel Functions 12/4/2019

'This program calculates the Bessel functions of the first kind provided an argument x and the order of the polynomials to be calculated.
'The program uses the original formula for bessel functions and also the recurrence relation.

'Welcome form
Public Class frmWelcome
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    End Sub

    'Subrutine for the start button
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles btnStart.Click
        frmCalculator.Show()
        Me.Hide()
    End Sub

    'Subrutine for the exit button
    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles btnExit.Click
        MsgBox("Thank You!" + vbNewLine + "For more information email luis.flores8@upr.edu",
               MsgBoxStyle.OkOnly Or MsgBoxStyle.Information,
               "Exit Application")
        Me.Close()
        Application.Exit()
    End Sub
End Class
```

```
'Luis A. Flores Carrubio (841-10-2500)
'University of Puerto Rico Rio Piedras
'Physics Department FISI6401
'Bessel Functions 12/4/2019

'This program calculates the Bessel functions of the first kind provided an argument x and the order of the polynomials to
be calculated.
'The program uses the original formula for bessel functions and also the recurrence relation.


Public Class frmCalculator
    'Variable used for the stop button
    Dim exitLoop As Boolean = False

    'Subrutine for the exit button
    Private Sub btnExit_Click(sender As Object, e As EventArgs) Handles btnExit.Click
        MsgBox("Thank You!" + vbNewLine + "For more information email luis.flores8@upr.edu",
               MsgBoxStyle.OkOnly Or MsgBoxStyle.Information,
               "Exit Application")
        Me.Close()
        Application.Exit()
    End Sub

    'Subrutine for the calculate button
    'This is the main part of the program, all formulas and calculations are here.
    Private Sub btnCalculate_Click(sender As Object, e As EventArgs) Handles btnCalculate.Click
        Dim lowerX, UpperX, Bessel, BesselRec, Jx, Jy As Decimal 'Decimal data type was used instead of Double as it is more
precise
        Dim lowerN, UpperN As Integer
        Dim mValue As Int64


        Try 'Try-Catch statement for validating the user's entered parameters
            lowerX = CDec(txtx.Text)    'lowerX is the minimum value of the argument x
            UpperX = CDec(txtx2.Text)   'UpperX is the maxium value of the argument x
            lowerN = CInt(txtn.Text)    'lowerN is the lowest order n polynomial
            UpperN = CInt(txtn2.Text)   'UpperN is the highest order n polynomial
            mValue = CInt(txtm.Text)    'mValue holds the value of the index of the summation
```

```
        Catch ex As Exception
            MessageBox.Show(ex.Message + vbNewLine + "Try using different values for the parameters.", "Input Error!",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
            txtm.Clear()
            txtn.Clear()
            txtn2.Clear()
            txtResults.Clear()
            txtx.Clear()
            txtx2.Clear()
            Exit Sub
        End Try

        If lowerN Or UpperN < 0 And lowerN Or UpperN > 15 Then    'Validating user input
            MsgBox("The values for the parameter alpha cannot be negative numbers or the values chosen might result in an
overflow.",
               MsgBoxStyle.OkOnly Or MsgBoxStyle.Exclamation,
               "Attention")
            txtn.Clear()
            txtn2.Clear()
            Exit Sub
        End If

        If mValue > 15 Or mValue < 0 Then   'Validating user input
            MsgBox("The value entered for parameter M might result in an overflow or it cannot be a negative number." +
vbNewLine + "Try a different value.",
             MsgBoxStyle.OkOnly Or MsgBoxStyle.Information,
                  "Attention")
            Exit Sub
        End If

        'Creating the table in which the results will display
        txtResults.Clear()
        txtResults.AppendText(vbTab)
        txtRec.Clear()
        txtRec.AppendText(vbTab)
        For n As Int16 = lowerN To UpperN Step 1
            txtResults.AppendText("n=" + n.ToString + vbTab + vbTab)
            txtRec.AppendText("n=" + n.ToString + vbTab + vbTab)
        Next
```

```vb
        'Loop in which the Bessel function is calculated is the the two formulas Bessel and BesselRec
        For x As Decimal = lowerX To UpperX Step 0.1
            Application.DoEvents() 'Stop button
            If exitLoop = True Then Exit For 'Stop button
            txtResults.AppendText(vbNewLine + "x=" + x.ToString + vbTab)
            txtRec.AppendText(vbNewLine + "x=" + x.ToString + vbTab)

            For n As Int16 = lowerN To UpperN Step 1
                Bessel = Math.Pow(x, n) * Sum(x, n, mValue) 'Calculating the original bessel function value given the
parameters
                If n = lowerN Then
                    Jx = Bessel  'J0 value corresponding the the first polynomial of the bessel function
                    txtRec.AppendText(Strings.Format(Bessel, "e") + vbTab)
                End If

                If n = lowerN + 1 Then
                    Jy = Bessel 'J1 value corresponding to the second polynomial of the bessel function
                    txtRec.AppendText(Strings.Format(Bessel, "e") + vbTab)
                End If

                If n > lowerN + 1 And x <> 0 Then 'Calculate the values of the bessel function using recurrence relation
                    BesselRec = ((2 * n / x) * Jy) - Jx 'Recurrence formula
                    'Swapping values
                    Jx = Jy
                    Jy = BesselRec
                    txtRec.AppendText(Strings.Format(BesselRec, "e") + vbTab) 'Appending results to the lower textbox area
                End If
                txtResults.AppendText(Strings.Format(Bessel, "e") + vbTab) 'Appending results to the upper textbox area
            Next
        Next
        exitLoop = False
    End Sub

    'Function for calculating the summation part involved in the bessel function
    'Returns the value of the sum given the argument x, order of polynomials n, and the maximum value of the sum index
    Public Function Sum(x As Decimal, n As Int16, mValue As Integer) As Decimal
        Dim valueSum As Decimal = 0 'Stores the summation value
        For m As Int64 = 0 To mValue Step 1
         valueSum += (Math.Pow(-1, m) * Math.Pow(x, (2 * m))) / (Math.Pow(2, (2 * m + n)) * Factorial(m) * Factorial(n + m))
        Next
        Return valueSum
    End Function
```

```
    'Function for calculating the factorial of a given number
    'The factorial is used since the gamma function can be reduced to a shifted factorial function
    Public Function Factorial(n As Decimal) As Decimal
        If n = 0 Then
            Return 1
        Else
            Try 'Statements are inside a try-catch in case an overflow occurs
                Return n * Factorial(n - 1)
            Catch ex As Exception
                MessageBox.Show(ex.Message + vbNewLine + "Values of M or alpha resulted in an overflow.", "Overflow!",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
                Exit Function
            End Try
        End If
    End Function

    'Subrutine for the stop button
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles btnStop.Click
        exitLoop = True
    End Sub

    'Subrutine for the clear button
    'Clears all strings in the controls
    Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
        txtm.Clear()
        txtn.Clear()
        txtn2.Clear()
        txtResults.Clear()
        txtRec.Clear()
        txtx.Clear()
        txtx2.Clear()
    End Sub

    Private Sub frmCalculator_Load(sender As Object, e As EventArgs) Handles MyBase.Load

    End Sub
End Class
```