1. Homework Problem 1: Numerical Integration using Trapezoidal Rule [20 points]

Write a program in C which computes the Numerical Integration of the following function using Trapezoidal Rule.

\[
\int_{-1}^{1} f(x) \, dx, \text{ where } f(x) = 2\sqrt{1-x^2}
\]

Numerical Integration means numerical evaluation of integral

\[
J = \int_{a}^{b} f(x) \, dx
\]

where \(a\) and \(b\) are given and \(f\) is a function given analytically by a formula or empirically by a table of values. Geometrically, \(J\) is the area under the curve of \(f\) between \(a\) and \(b\).

Numerical integration methods are obtained by approximating the integrand \(f\) by functions that can be easily be integrated. The Trapezoidal Rule is one such method, which is quite accurate. We subdivide the interval of integration \(a = x = b\) into \(n\) subintervals of equal length \(h = (b - a)/n\) and then find the total area under the curve \(f\) between \(a\) and \(b\) by summing the individual areas of the trapezoids formed by each of the subintervals.

The resultant formula is:

\[
J = \int_{a}^{b} f(x) \, dx = h \left[ \frac{1}{2} f(a) + f(x_1) + f(x_2) + \cdots + f(x_{n-1}) + \frac{1}{2} f(b) \right]
\]

\[
= \frac{h}{2} [f(a) + f(b)] + h [f(x_1) + f(x_2) + \cdots + f(x_{n-1})]
\]

where \(h = (b - a)/n\) is the subinterval, \(x_0 = x_0, f + h, a = x_0\) and \(b = x_n\).
In your C code, to compute the given function \( f(x) = 2\sqrt{1 - x^2} \) in the above formula, take the following into account:

1. Data type for the variables should be `float`.
2. To compute the square root, use the function `sqrt()` which is defined in the header file `math.h`. For example, to find a square root of any variable \( x \) and store it in any variable \( y \), the C code should be \( y = \text{sqrt}(x) \). Include the following header files on top of your code:
   ```c
   #include <stdio.h>
   #include <math.h>
   ```
3. You will need to include an extra option when compiling programs that use functions from `math.h`. Include the option, “-lm” when compiling. This option tells the linker where to find the libraries required to use the math functions.

When executed, your program should look like this:

Please enter the number of subdivisions \( n \):
The approximate integration by Trapezoidal Rule is:

You should implement the above code using loops. Compile your program and run it using \( n = 100, 1000, \) and \( 10000 \) as inputs from the keyboard.

You should submit `trapezoidal.c`, `trapezoidal.txt` and `trapezoidal.script` for this problem.

### 2. Extra Credit Problem: Monte Carlo Integration

In the Monte Carlo Integration method, we pick random numbers evenly distributed between the given bounds and find the value of the function at each of these points. We use these values to compute the average value of the function. We then multiply this average by the distance of the integral to get the area under the curve.

If we pick \( N \) random points, then the Monte Carlo Integration of function \( f(x) \) will be:

\[
J = \frac{V}{N} \sum_{i=1}^{N} f(x_i),
\]

where \( V \) is the distance the integral is over (\( b-a \) in this case).

When executed, your program should look like this:

Please enter the number of subdivisions \( n \):
The approximate integration by Monte Carlo Method is:

Implement the C code using loops. Compile your program and run it using \( n = 100, 1000, \) and \( 10000 \) as inputs from the keyboard.

**HINT:**

For generating the initial random number, you have to use a random number generator, which is provided by the C standard function `rand()`. This function generates a random number of type `int` in the range of \( 0 \) to \( 32767 \). This function is provided in the header file `stdlib.h`.

In summary, use the following code fragments to generate the random number for the game:

1. Include these header files at the beginning of your program:
   ```c
   #include <stdio.h>
   ```
#include <stdlib.h>
#include <math.h>

2. Include the following lines at the beginning of your main function:

/* generate the random number in the range 0 to (urange -1) */
randomNumber = rand() % urange;

Here, urange specifies the upper bound of the range in which the random number will be generated, and randomNumber is the integer variable which is assigned the generated random number.

3 What to turn in

Submission for these files will be similar to last week’s assignment. The only difference is that you need to create a directory called hw2/. Put all the files listed above in that directory and run the /ecelib/bin/turnin command to submit your homework. You should turn in the following files:

trapezoidal.c
trapezoidal.txt
trapezoidal.script
monte_carlo.c
monte_carlo.txt
monte_carlo.script

You can use the following links for references:

http://people.hofstra.edu/faculty/Stefan_Waner/RealWorld/integral/numint.html
http://mathworld.wolfram.com/MonteCarloIntegration.html