Assignment 1 DESIGN.pdf

Description of Program:

This bash script produces plots of the given C program, which generates a predetermined amount of random two-dimensional points and graphs them. One plot demonstrates the number of points that fall within a quarter circle inscribed within a square and denotes them with the color blue, and red otherwise. The other plot displays the error values of the estimated π value as the number of iterations increases.

Files to be included in directory "asgn1":

1. plot.sh
    - Source file containing bash script that plots given data points and exports them as pdfs
2. monte_carlo.c
    - Generates X random (x,y) data points inside a square size 1,1, and measures the number of points less than or equal to 1 unit distance away from the origin (fall within the inscribed circular quadrant). The estimated π value will be outputted and a value of 1 will be assigned to the point if it falls inside the circle, and 0 otherwise.
3. Makefile
    - Supports the compilation of the Monte Carlo program
4. README.md
    - Proper explanation of how to use script file and *Makefile*. Also describes any command-line options in the program.
5. DESIGN.pdf
    - Explains the thought process behind the program with enough information to easily replicate the implementation. Must be in PDF format.
6. WRITEUP.pdf
    - PDF file created with LaTeX that includes plots created using bash script with an explanation of UNIX commands.
7. WRITEUP.tex
    - Latex file that generates WRITEUP.pdf

Pseudocode / Structure:

Rebuild the monte_carlo executable

Run monte_carlo up to X amount of data points (currently set to 100) and preserve all lines except line 1, then write to a new data file *A (containing iteration number, point coordinates, estimated π value, and whether they fall within the circle)*

Search through the column containing whether the points falls within the circle and check if it equals 1 → append X and Y coordinates of that line to new data file *B* (points inside the circle)

Search through the column containing whether the points falls within the circle and check if it equals 0 → append X and Y coordinates of that line to new data file *C* (points outside the circle)

Send to Gnuplot
Set output as PDF and name it "monte_carlo"
Label the x-axis "x" and y-axis "y"
Create a square (side length 1) and a quarter circle inside the square (radius of 1) and plot them
Plot points that fall inside the circle (B) in blue
Plot points that fall outside the circle (C) in red

For i in 4:

> Run monte_carlo and generate 70,000 points (with random seed), preserving all but the first line, then subtract accurate pi value from each estimated pi value, then save to a new data file

Send to Gnuplot
Set output as PDF and name it "error"
Label the x-axis "iterations" and y-axis "error"
Set the graph title as "Monte Carlo Error Estimation"
Set the range of the y-axis from -1 to 1
Set the x-axis to increase logarithmically with base 4 and give it a range from 1 to 4^8
Plot the four data files, making sure to assign each line a different color

Credit:

- The provided *plot.sh* consisted of bash script code for *sincos.c*, mainly syntax and initialization. I used some of this to set up my monte_carlo.c script. I also included some lines of code regarding setting up and displaying plots using gnuplot, using them as nothing more than formatting tools.

- In order to subtract π from the estimated π values in the Monte Carlo output data file, I used a version of Professor Long's code which he posted into the CSE 13S Discord server. I personally implemented the code that would perform this arithmetic on 4 different data files using different seeds.