Name: ______________________________

Date: ______________________________

Question:	1	2	3	4	Total
Points:	5	10	5	40	60
Score:					

1. (5 points) You selected Monte Carlo method to calculate a 7-dimensional integral. You generated $N = 10^6$ random points per each of $m = 128$ runs of your algorithm and found the result with the standard deviation of 1.28×10^{-2}. However, you'd like to achieve the standard deviation of the result of order 1.0×10^{-3}, i.e. about 10 times smaller. What do you need to do?

   - ☐ Increase the number of runs, m, by a factor ~ 10
   - ☐ Increase the number of runs, m, by a factor $\sim 10^2$
   - ☐ Increase the number of random points per run, N, by a factor ~ 10
   - ☐ Increase the number of random points per run, N, by a factor $\sim 10^2$
   - ☐ increase the number of the dimensions in the integrand to 10 by introdicing three "dummy" dimensions
   - ☐ All of the above
   - ☐ None of the above

   Please explain:

   ____

   ____

2. (10 points) You need to design a linear congruental random number generator that is supposed to produce unsigned random integers on a microchip that uses 10 bit integers. What is the largest possible period of your generator?

   ____

   Please explain:

   ____

   ____

3. (5 points) What are some of desired properties of a random number generator?
   - ☐ High performance
   - ☐ Long period
   - ☐ Repeatability
   - ☐ Secrecy of the algorithm
   - ☐ All of the above
   - ☐ None of the above

4. (40 points) Use Monte Carlo integration to calculate the volume of n-dimensional sphere of unit radius, V_n, for $n = 10, 12, 14,$ and 16. Compare your results with the analytic expression for the volume:

$$V_n = \frac{\pi^{\left(\frac{n}{2}\right)}}{\left(\frac{n}{2}\right)!} \quad \text{(for even n)}. \tag{1}$$

   Hint 1: n-dimensional sphere of unit radius is the set of points which are at distance less than one from a central point. The unit sphere centered at the origin is described by the following relation:

$$x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2 \leq 1$$

   Hint 2: generate at least 10^8 n-dimensional random points.

   Hint 3: Use functions `pow(x, n)` defined in `math.h` and `gsl_sf_fact(i)` defined in `gsl_sf_gamma.h` to calculate Eq. (1).

   Provide printouts of your (formatted) C code, your makefile, and (nicely formatted) output of your program.