CSC1321 Programming in C++

Programming Assignment: Approximation of π

Objectives

- Be familiar with looping structure and random number generator.
- Be able to solve problems using looping structures

Key Ideas

- Random number generation: rand( )
- Looping: while, do-while, and for loops

Problem Description

π is a mathematical constant that is the ratio of any circle's circumference to its diameter. π is approximately equal to 3.14 in the usual decimal notation. Many formulas in mathematics, science, and engineering involve π, which makes it one of the most important mathematical constants. $\pi(3.14159265\ldots)$ is an irrational number, which means that its value cannot be expressed exactly as a fraction having integers in both the numerator and denominator. Consequently, its decimal representation never ends or repeats. (from wikipedia.com)

The following describes a method of approximating the value of π by throwing darts. Suppose that a dart player throws a number of darts at a circular target inscribed into a square and all the darts hit inside the square. Figure 1 shows an imaginary distribution of darts on the target after 150 darts have been thrown. To simplify the problem, assume that the chances to hit anywhere inside the square are equal. If a dart hits inside the circle, it is counted as a hit, otherwise a miss. It is observed that the chance (probability) to hit inside the circle is directly proportional to the ratio of the area of the circle to the area of the square.

A mathematical formula is used to express the relationship as:

$$\frac{H}{T} = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4}$$

where H is the number of hits; T is the total number of darts thrown; R is the radius of the circle. Then π can be expressed as:

$$\pi = 4\frac{H}{T}$$

The steps to approximate the value of π:

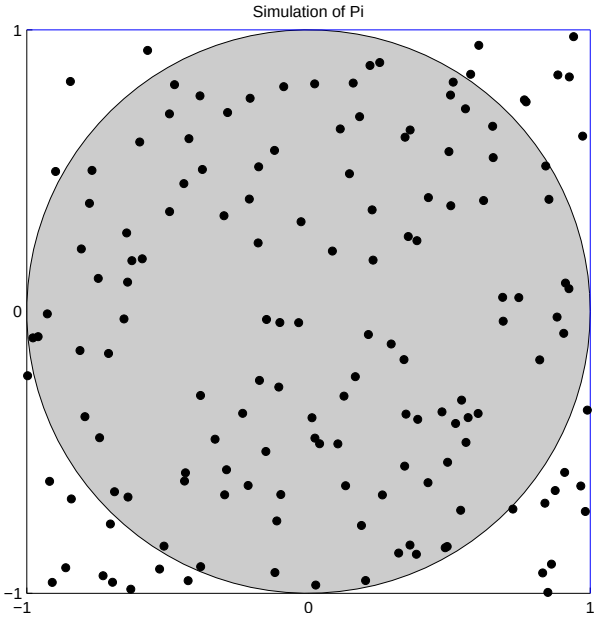

Figure 1: Distribution of Darts on a Target

1. Set up a coordinate system with its origin at the center of the circle.
2. Generate a pair of x and y coordinates of darts that represents the location of a dart hit on the target using the C++ built-in random number generator function called *rand*(). It generate a random integer number between 0 and RAND_MAX. Constant RAND_MAX defines the maximum possible random number that rand() could generates. You need to scale the random numbers in a range between 0 and the radius of the circular target (exclusive).
3. Calculate the distance between the darts and the origin of the coordinate system with the formula below:
$$d = \sqrt{x^2 + y^2}$$
4. Assume that the radius R of the circular target is 1. If d is less than R, count as a hit, otherwise a miss.
5. Count the number of hits, given the total number of darts thrown.
6. Calculate π using the preceding formula $\pi = 4\frac{H}{T}$.

Note that darts could hit any of the four quadrants of the coordinate system. The pairs of x and y could be either positive or negative. However, rand() function generates only positive numbers. This "sign problem" of coordinates is corrected automatically when you calculate the distance between the origin and a dart point because x and y are squared. The above procedure still gives correct result.

You need to try different number of the darts thrown and see if the approximation to the value of π improves. Turn in the printout of your program and test results. If you present your result with a graphical chart like Figure 1 (MS Excel might be a good tool for this), you will receive 20% extra points.