General

This program estimates the value of pi using the Monte Carlo method. It generates a specified number of random points within a unit square and determines how many fall within the unit circle inscribed in that square. By comparing the ratio of points inside the circle to the total number of points, it estimates the value of pi.

Usage

To compile and run the program, execute the following commands:

```
make n=9999
```

The program will estimate the value of pi using the Monte Carlo method with 9999 random points. The output will display the estimated value of pi along with the elapsed time for computation.

Implementation

- The program includes necessary header files: stdio.h, stdlib.h, time.h, and pthread.h.
- It declares a global variable 'num_points_circle' to keep track of the number of points inside the circle.
- A function 'runner' is defined to perform the Monte Carlo simulation.
- A separate thread is created to create the points and to count the number of points inside the circle.
- After joining the thread, it calculates the estimated value of pi using the formula: pi = 4 * (num_points_circle / n).

Analysis

number of random points	Approximated PI
9	3.111111
99	3.151515
999	3.131131
9999	3.171117
99999	3.141511
999999	3.141663
9999999	3.141130
99999999	3.141698

From the table, we can see that the estimated value of pi gets closer to the actual value as the number of random points increases. The Monte Carlo method is a probabilistic algorithm, and the accuracy of the estimation depends on the number of random points generated. The more points generated, the more accurate the estimation will be.

Screenshots

The following screenshots illustrate the program's execution and output for estimating the value of pi using the Monte Carlo method with 9999 random points.