

prog_sequencial.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  #include <omp.h> // Cabeçalho necessário para omp_get_wtime()
5
6  // Definição global do número de passos para consistência
7  const long NUM_PASSOS = 100000000;
8
9
10 void pi_sequencial() {
11     long pontos_no_circulo = 0;
12     unsigned int seed = 12345; // Semente fixa para repetibilidade
13
14     for (long i = 0; i < NUM_PASSOS; i++) {
15         double x = (double)rand_r(&seed) / RAND_MAX * 2.0 - 1.0;
16         double y = (double)rand_r(&seed) / RAND_MAX * 2.0 - 1.0;
17         if (x * x + y * y < 1.0) {
18             pontos_no_circulo++;
19         }
20     }
21     double pi = 4.0 * pontos_no_circulo / NUM_PASSOS;
22     printf("Sequencial: pi = %f\n", pi);
23 }
24
25
26 int main() {
27     double start_time, end_time;
28
29     printf("Iniciando análise de desempenho para %ld passos.\n", NUM_PASSOS);
30
31     // Teste Sequencial
32     start_time = omp_get_wtime();
33     pi_sequencial();
34     end_time = omp_get_wtime();
35     double tempo_sequencial = end_time - start_time;
36     printf("Tempo Sequencial: %f segundos\n", tempo_sequencial);
37
38     return 0;
39 }
```