CSE4001 – Lab Assessment 1

17BCI0113 – Namit Nathwani
Prof. Jaisakthi S M – L55+L56

Question: Implement a parallel program to study the performance of the parallel for using OpenMP and compare the result with serial code.

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <omp.h>

using namespace std;

void doSum(double *A, int n, int threads) {
     omp_set_dynamic(0);
     double sum = 0;

     #pragma omp parallel for num_threads(threads) reduction(+:
sum)
          for(int i = 0; i < n; i++)
               sum += A[i];

     cout << "Sum = " << sum << endl;
}

void normalDoSum(double *A, int n) {
     double sum = 0;

     for(int i = 0; i < n; i++)
```

```
            sum += A[i];

      cout << "Sum = " << sum << endl;
}

int main() {
      int i, n, threads;
      double *A, dtime, sum;

      n = 100000000;
      A = (double*) malloc(sizeof(double) * n);

      for(i = 0; i < n; i++)
            A[i] = 1.0 * rand()/RAND_MAX;

      cout << "n = " << n << endl;

      cout << "Dynamic (8 threads):" << endl;
      threads = 8;
      dtime = omp_get_wtime();
      doSum(A, n, threads);
      dtime = omp_get_wtime() - dtime;

      cout << "Threads = " << threads << endl;
      cout << "Time Taken = " << dtime << "s\n" << endl;



      cout << "\nSerial:" << endl;
      dtime = omp_get_wtime();
      normalDoSum(A, n);
```

```
	dtime = omp_get_wtime() - dtime;


	cout << "Time Taken = " << dtime << "s\n" << endl;
}
```

Output:

```
17bci0113@sjt418scs045:~/Documents/Lab Assessment 1$ gcc 17BCI0113-Assessment1.cpp -fopenmp -lstdc++
17bci0113@sjt418scs045:~/Documents/Lab Assessment 1$ ./a.out
n = 100000000
Dynamic (8 threads):
Sum = 5.00011e+07
Threads = 8
Time Taken = 0.0712843s


Serial:
Sum = 5.00011e+07
Time Taken = 0.246799s
```