Nikhil Rajendra Danapgol
HPC Assignment 1
2019BTECCS00036

Q1. Program in OpenMP to print “Hello World”.

```
#include <omp.h>
#include <stdio.h>
#include <stdlib.h>
#include <bits/stdc++.h>

using namespace std;
// Hello World Program

int main(int argc, char *argv[])
{

#pragma omp parallel
    {
        printf("thread No. %d  Hello World\n", omp_get_thread_num());
    }
    return 0;
}
```

Output:

```
PS C:\HPCLAB> ./a.exe
thread No. 2  Hello World
thread No. 4  Hello World
thread No. 3  Hello World
thread No. 5  Hello World
thread No. 1  Hello World
thread No. 6  Hello World
thread No. 0  Hello World
thread No. 7  Hello World
PS C:\HPCLAB>
```

Q2. Find the squares of first 100 numbers followed by their sum. Compare the speed in sequential and parallel algorithm.

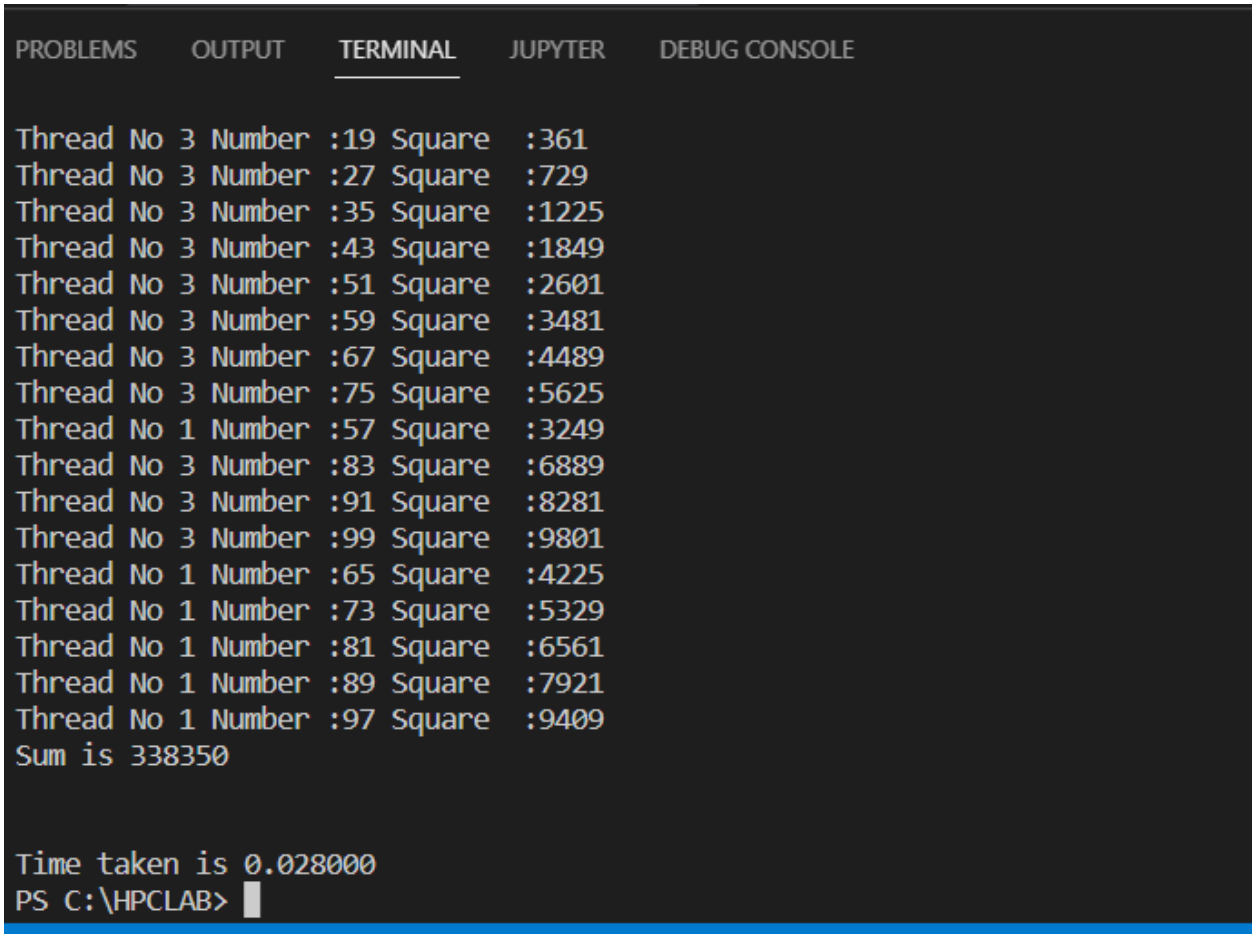
Parallel Approach:

HelloWorld.cpp

HelloWorld.cpp > main(int, char * [])

```
19  static int sum = 0;
20
21  // This is for parallel processing ...Calculates time for parallel processes
22
23  int main(int argc, char *argv[])
24  {
25
26      double itime, ftime, exec_time;
27      itime = omp_get_wtime();
28  #pragma omp parallel
29
30      for (int i = 1; i <= 100; i++)
31      {
32          if (i % 8 == omp_get_thread_num())
33          {
34              printf("Thread No %d Number :%d Square  :%d \n", omp_get_thread_num(), i, i * i);
35              sum += i * i;
36          }
37      }
38      printf("Sum is %d  ", sum);
39      cout << endl;
40      ftime = omp_get_wtime();
41      exec_time = (ftime - itime);
42      printf("\n\nTime taken is %f", exec_time);
43
44      return 0;
45  }
46
```

Output:



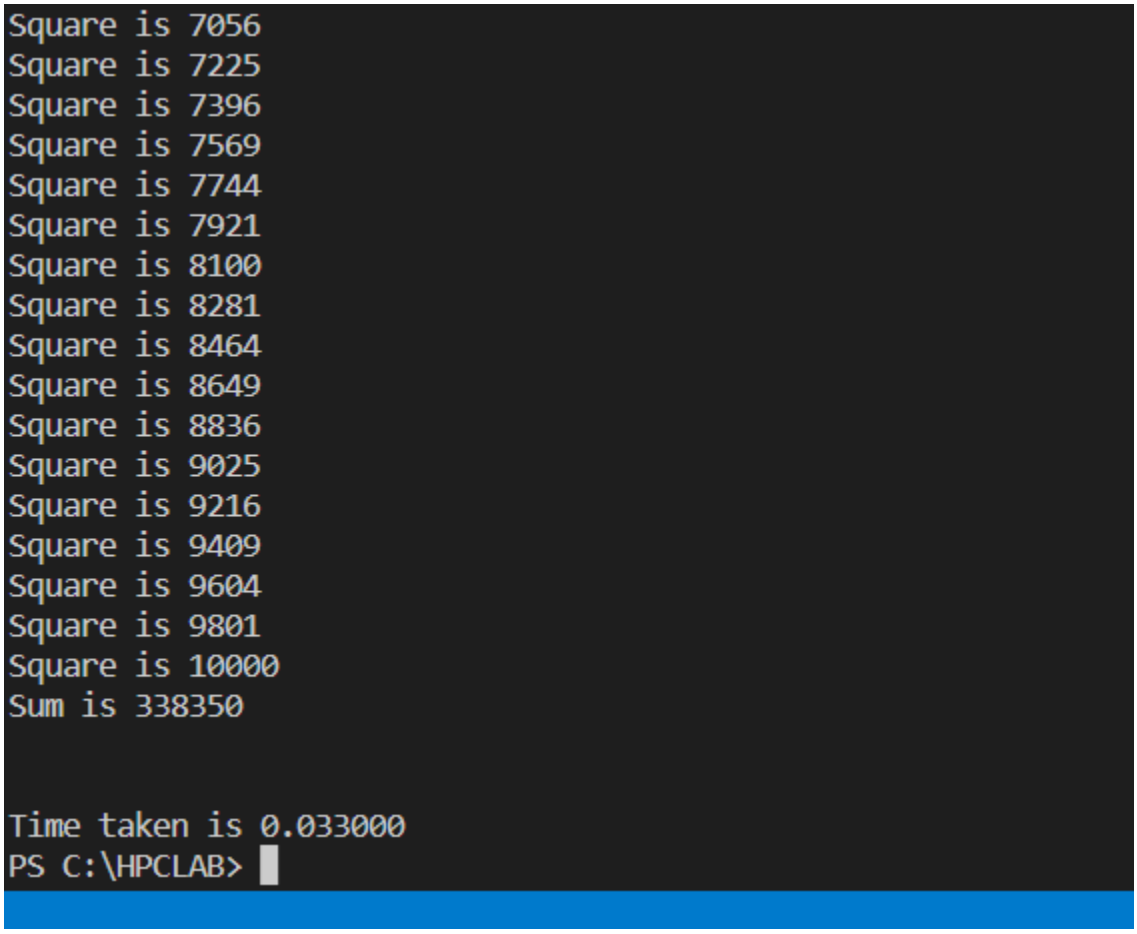
Sequential Approach:

```
int main()
{
    double itime, ftime, exec_time;
    itime = omp_get_wtime();
    // #pragma omp parallel

    for (int i = 1; i <= 100; i++)
    {
        // if (i % 8 == omp_get_thread_num())
        // {
        //  printf("Thread No %d Number :%d Square  :%d \n", omp_get_thread_num(), i, i * i);
        printf("Square is %d \n", i * i);
        sum += i * i;
        // }
    }
    printf("Sum is %d  ", sum);
    cout << endl;
    ftime = omp_get_wtime();
    exec_time = (ftime - itime);
    printf("\n\nTime taken is %f", exec_time);

    return 0;
}
```

Output:



Github Link:

https://github.com/nd22052000/HPC