Walchand College of Engineering, Sangli

Department of Computer Science & Engineering

Class : Final Year(Computer Science and Engineering)

Year : 2022-23

Semester 1

Course: High Performance Computing Lab

Name : Siddharth Mukkanawar

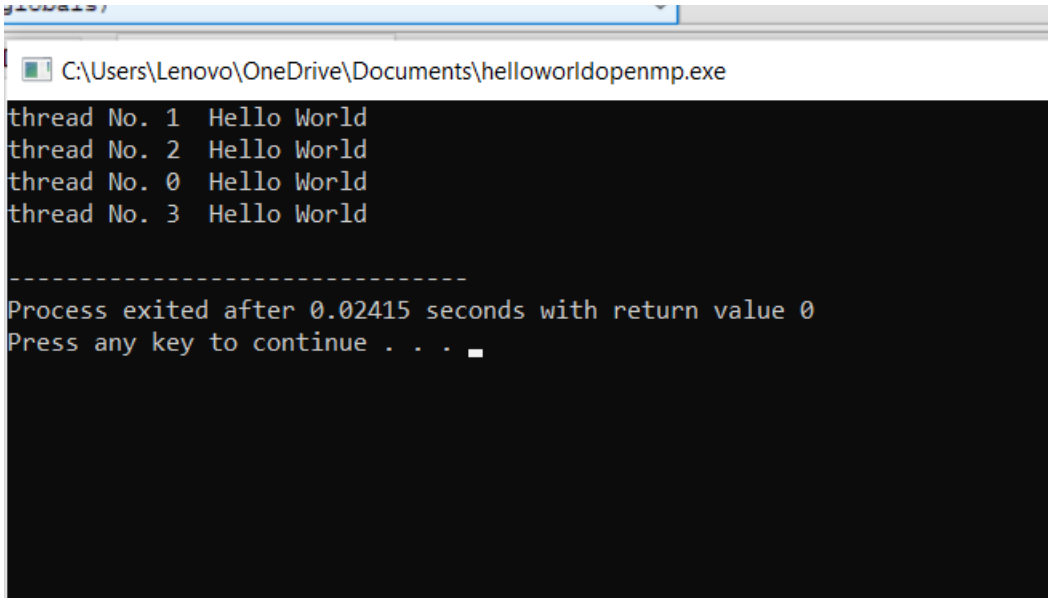
PRN: 2019BTECS00100

Batch: B4

Q1) Write a program to print Hello World using OpenMp

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

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

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

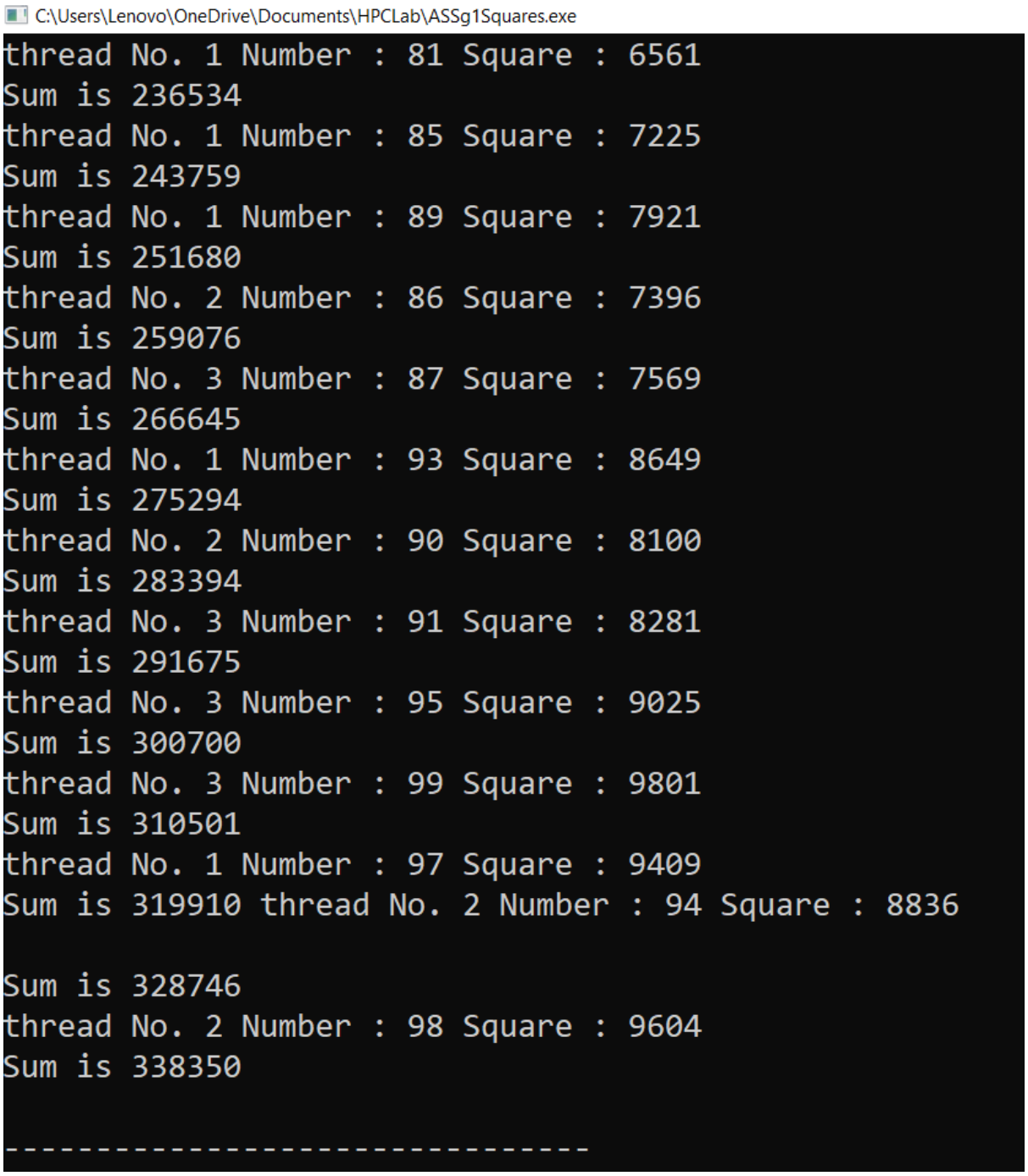
```
        }
    return 0;
}
```



Q 2) Write a program to print the sum of squares of numbers from 1 to hundred using OpenMp

```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>
#include<bits/stdc++.h>
using namespace std;



static int sum =0;
int main()
{
    #pragma omp parallel
```

```
  {


    for(int i=1; i<=100;i++)
    {
      if(i%4==omp_get_thread_num())
      {
        printf("thread No. %d Number : %d Square : %d\n", omp_get_thread_num(), i, i *
i);
        sum+=i*i;
        printf("Sum is %d ",sum);
        cout<<endl;


      }
    }
  }
  return 0;
}
```



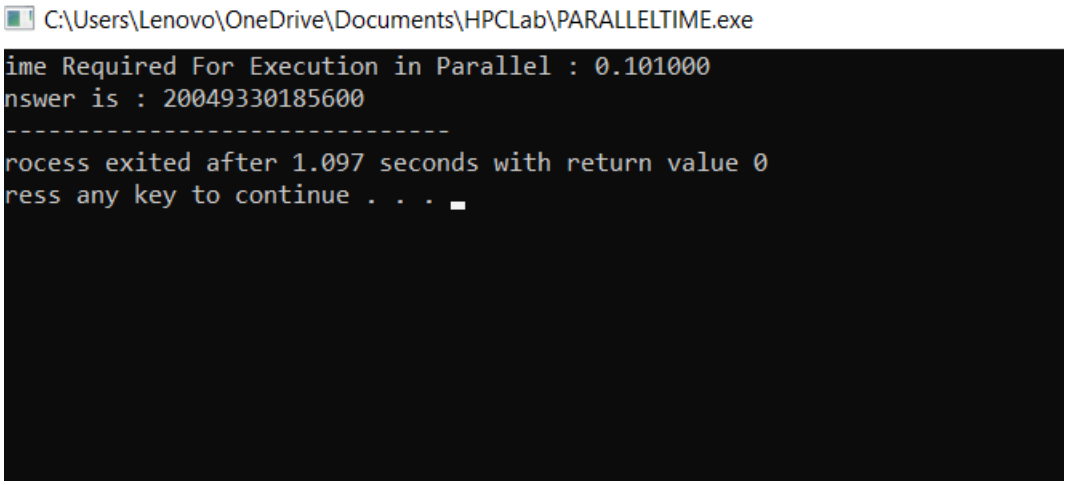
Q3) Write a programme to calculate the 5the speedup in parallel and sequential execution

1) Parallel execution:

```
#include<omp.h>

#include<stdio.h>

#include<stdlib.h>
```

```
#include<bits/stdc++.h>
using namespace std;


 int main(){
     long long sum = 0;

     double getInTime = omp_get_wtime();

     #pragma omp parallel for reduction(+ : sum)

     for(int i=1;i<=100000000;i++){
               sum += (i*i);
      }

     double getOutTime = omp_get_wtime();

     double exptTime = getOutTime - getInTime;


     printf("Time Required For Execution in Parallel : %f\n",exptTime);

     printf("Answer is : %lld",sum);

     return 0;
```

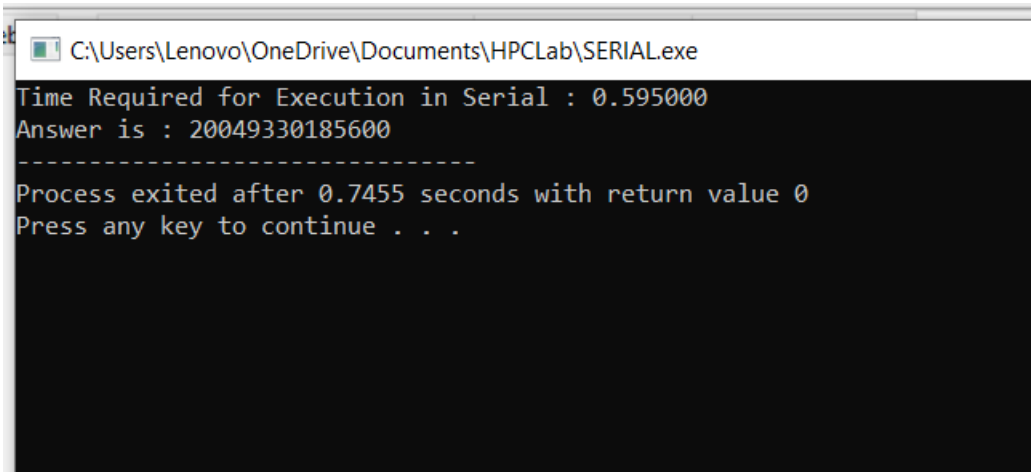
}



2) Sequential execution

```
#include<omp.h>
#include<stdio.h>
#include<stdlib.h>
#include<bits/stdc++.h>
using namespace std;


 int main(){

        long long sum = 0;

        double inTime = omp_get_wtime();


        int i;
        for(i=1;i<=100000000;i++){
                sum += (i*i);
         }


         double outTime = omp_get_wtime();

         double expcTime = outTime - inTime;
```

```
        printf("Time Required for Execution in Serial : %f\n",expcTime);
        printf("Answer is : %lld",sum);

        return 0;
}
```



Speedup =Sequential time/Parallel time

= 0.59500/0.10100

Speedup = 5.89

Github link: https://github.com/SiddharthM29/HPC_lab/tree/main/Assignment1