**Final Year B. Tech., Sem VI 2021-22**

**High Performance Computing Lab**

**Assignment submission**

**PRN No: 2019BTECS00070**

**Full name: Prathmesh Basaweshwar Killedar**

**Batch: B3**

**Assignment: 1**

Q1.Hello world

Code:-

#include<omp.h>

#include<bits/stdc++.h>

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

#pragma omp parallel

{

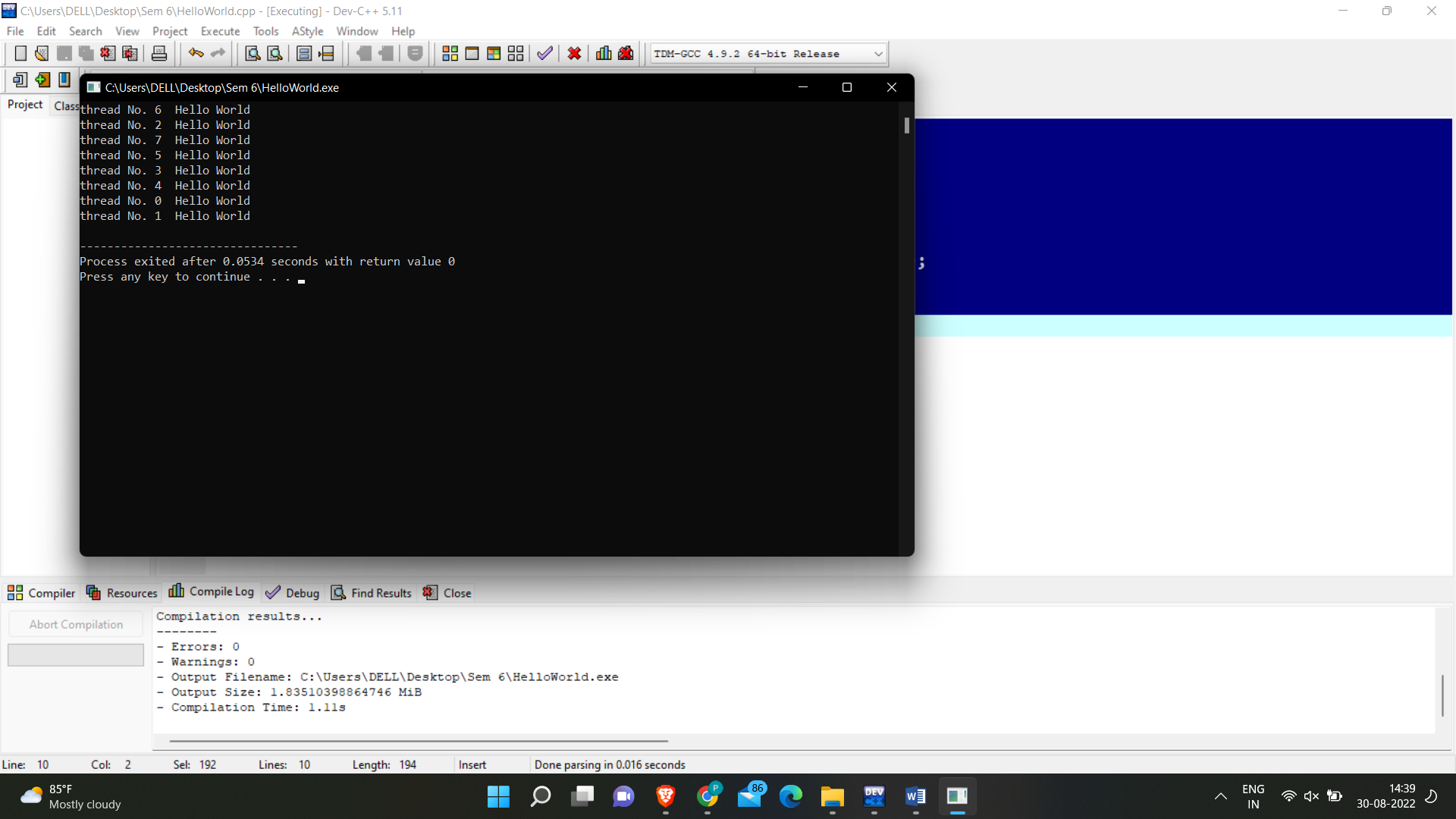
printf("thread No. %d Hello World\n", omp\_get\_thread\_num());

}

return 0;

}

Output:-



Q2.Squares

Code:-

#include<omp.h>

#include<bits/stdc++.h>

using namespace std;

static int sum =0;

int main()

{

#pragma omp parallel for

for(int i=1; i<=100;i++)

{

printf("thread No. %d Number : %d Square : %d\n", omp\_get\_thread\_num(), i, i \* i);

sum+=i\*i;

printf("Sum is %d \n",sum);

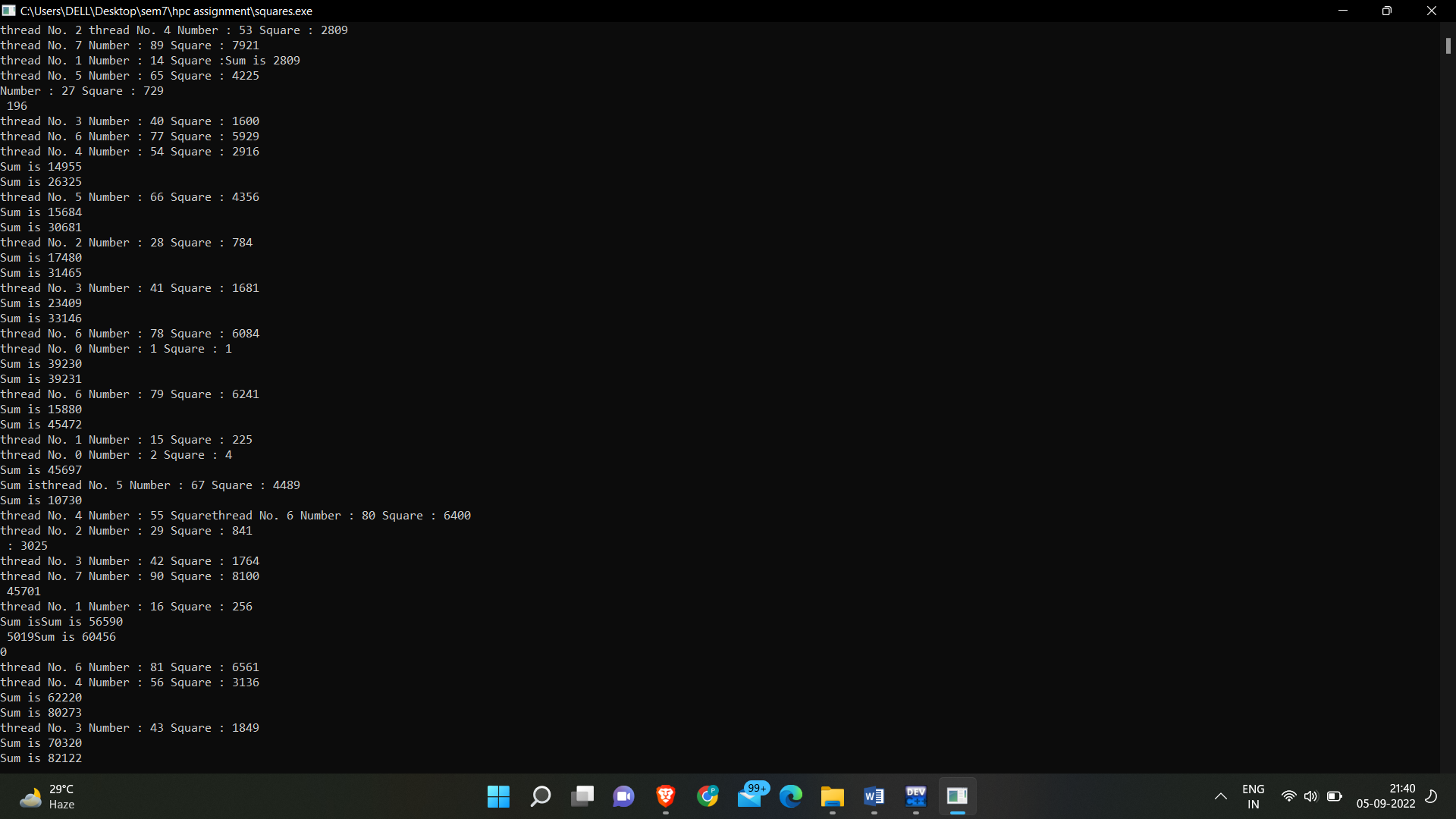
}

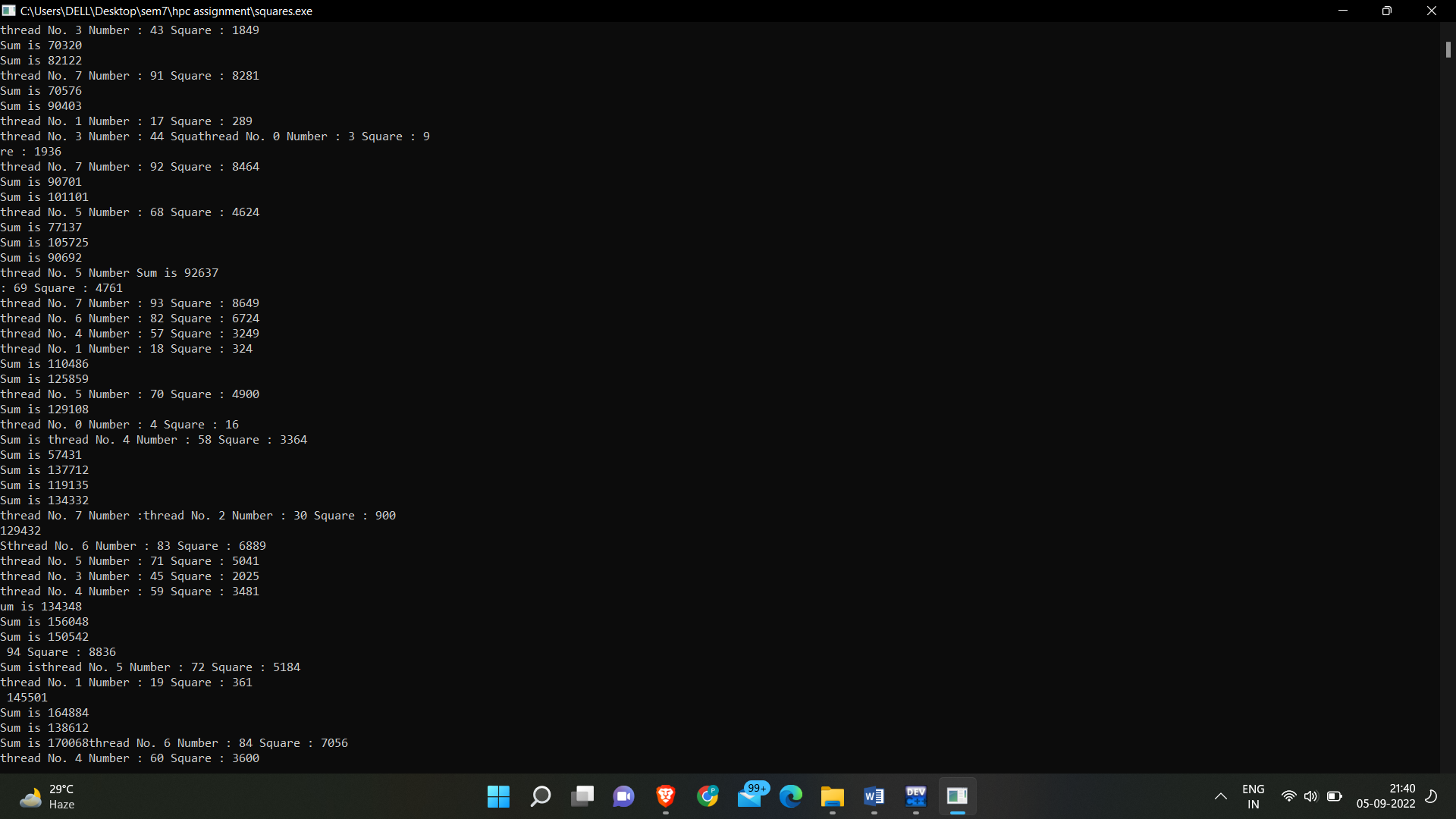
printf("%d",sum);

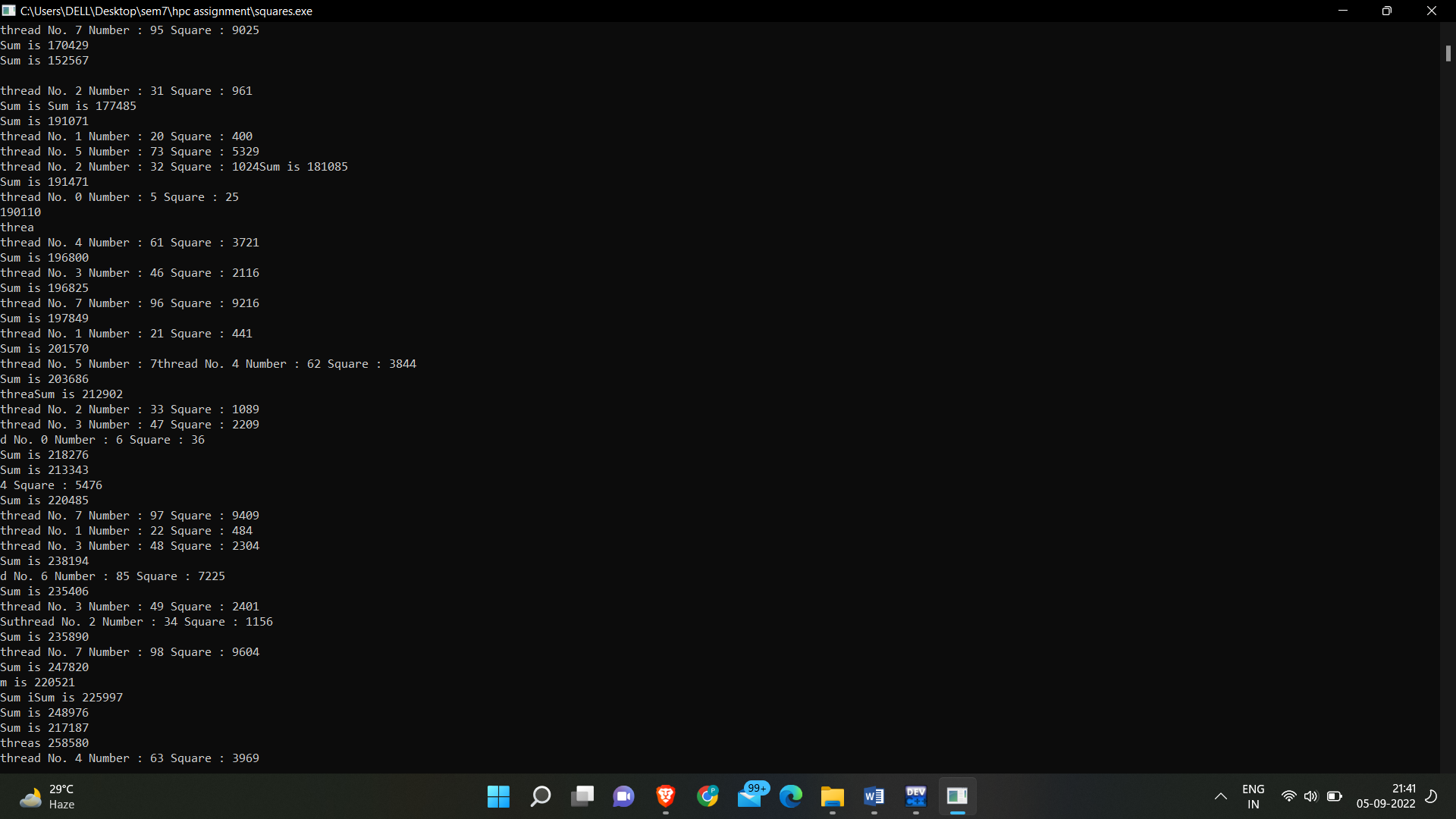
return 0;

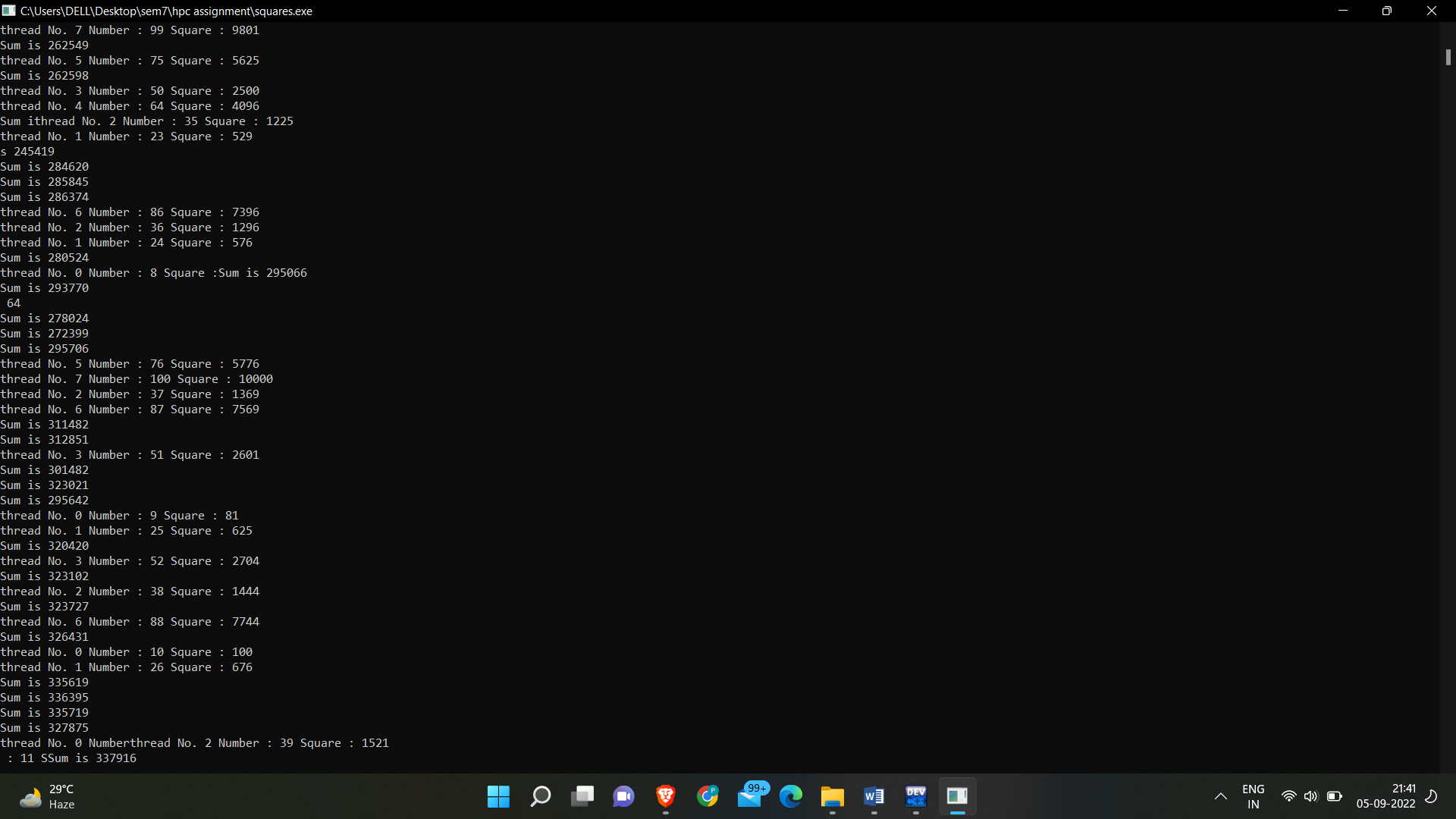
}

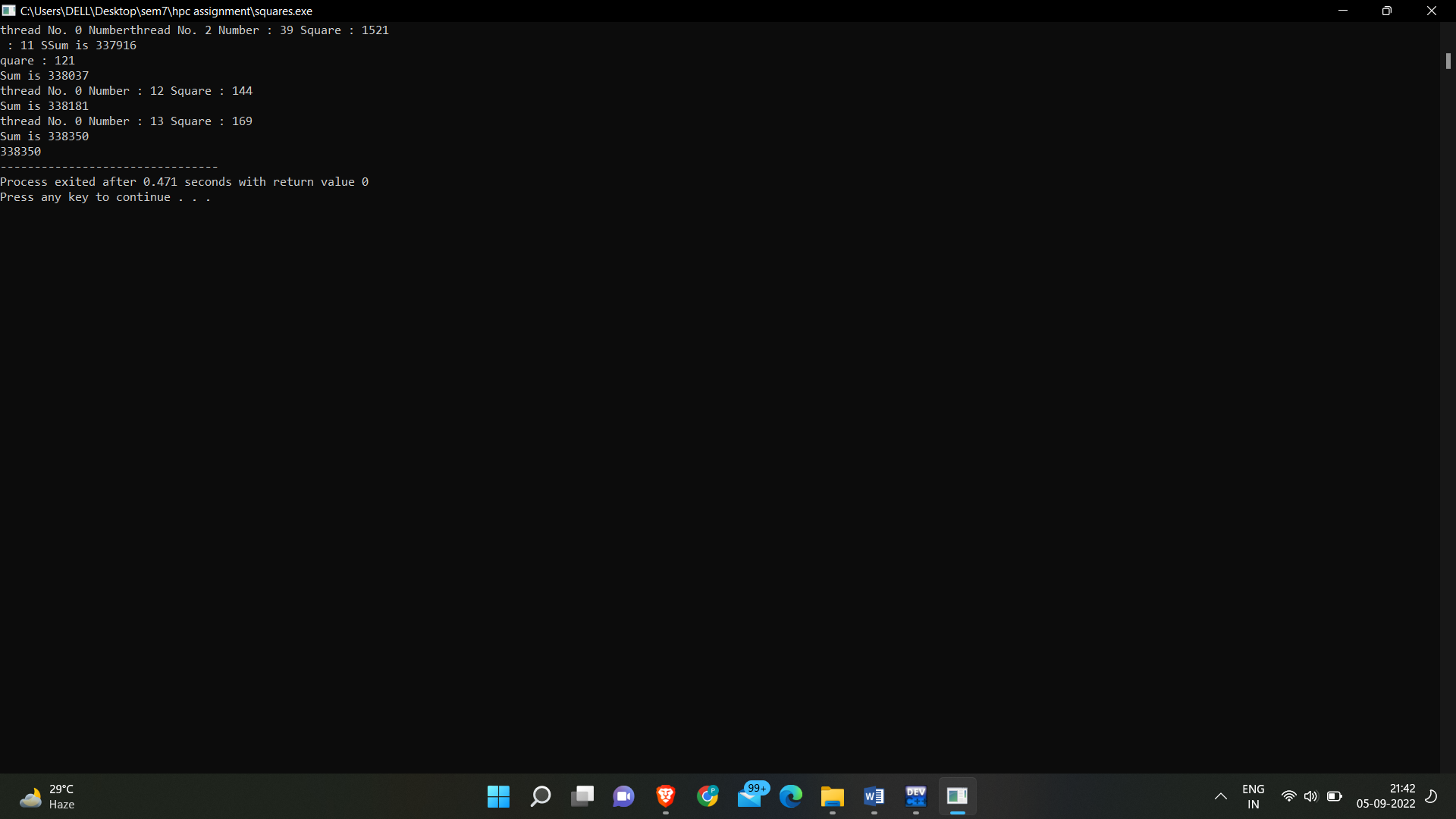
Output:-











Parallel Vs Serial:-

Parallel Code:-

#include<omp.h>

#include<bits/stdc++.h>

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;

}

Serial Code:-

#include<bits/stdc++.h>

#include<omp.h>

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;

}

Output:-

