NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA SURATHKAL

DEPARTMENT OF INFORMATION TECHNOLOGY

IT 301 Parallel Computing LAB 6

16th September 2020

Faculty: Dr. Geetha V and Mrs. Thanmayee

Roll no. : 181IT111

Bhagyashri Nilesh Bhamare.

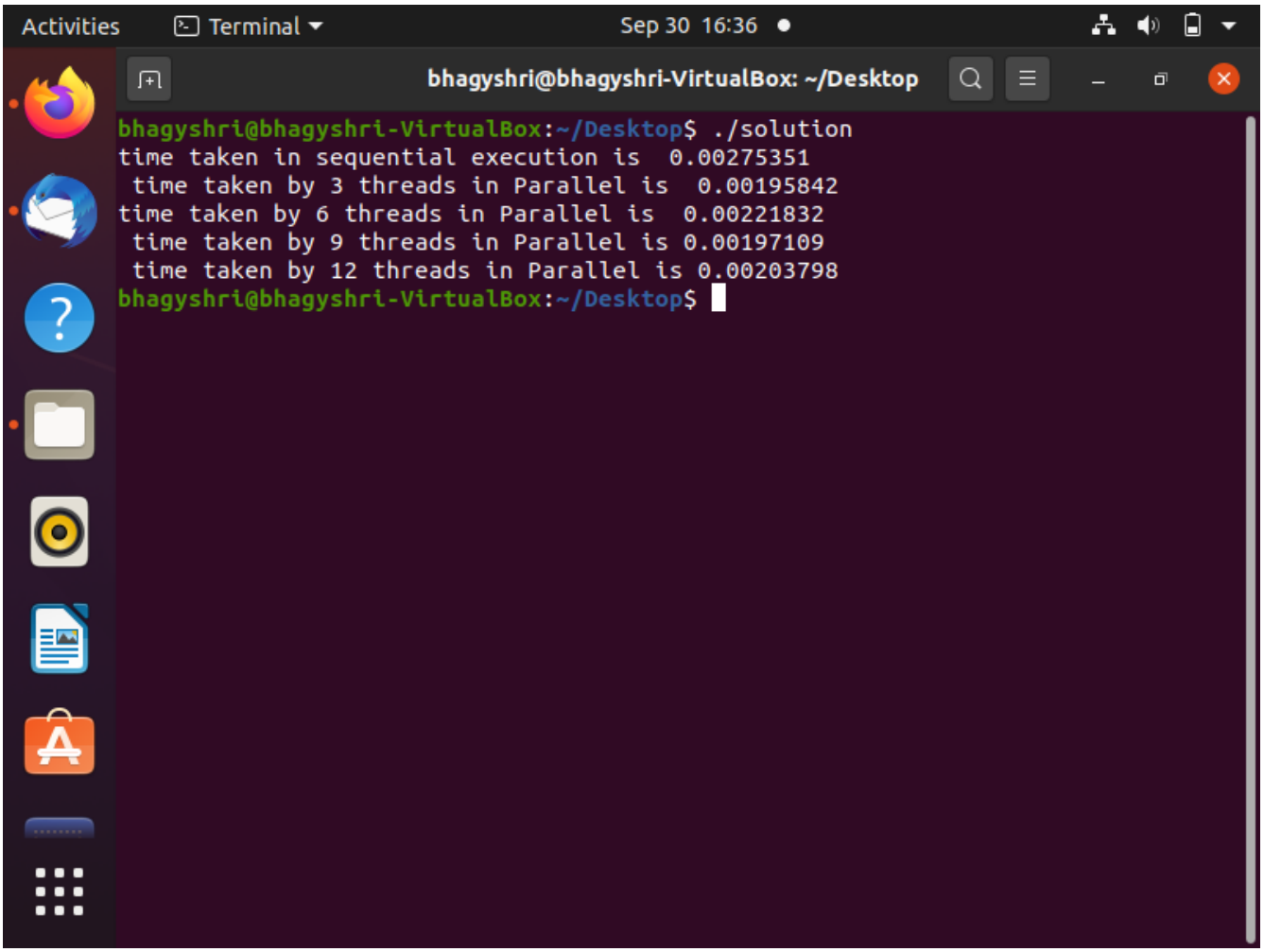
Code:

```
#include<bits/stdc++.h>
#include<omp.h>
using namespace std;
#define N 300
int mat[N*N*3],mat_gray[N*N],mat_yiq[3*N*N];
int main(){
freopen("KittenRGB.txt","r",stdin);
for(int i=0;i<3*N*N;i++){
cin>>mat[i];
}
// Sequential Execution
double start,end;
// RGB to GRAY scale
start=omp_get_wtime();
for(int i=0;i<300*300*3;i+=3){
int R=mat[i];
int G=mat[i+1];
int B=mat[i+2];
mat_gray[i]=(R*0.21)+(G*0.72)+(B*0.07);
mat_yiq[i]=(0.299*R)+(0.587*G)+(0.114*B);
mat_yiq[i+1]=(0.596*R)-(0.275*G)-(0.321*B);
mat_yiq[i+2]=(0.212*R)-(0.523*G)+(0.311*B);
}
end=omp_get_wtime();
cout<<"time taken in sequential execution is  "<<(end-start)<<'\n';
// Parallel Execution
// Number of threads = 3
start=omp_get_wtime();
#pragma omp parallel num_threads(3)
{
#pragma omp for
for(int i=0;i<N*N*3;i+=3){
```

```
int R=mat[i];
int G=mat[i+1];
int B=mat[i+2];
mat_gray[i]=(R*0.21)+(G*0.72)+(B*0.07);
mat_yiq[i]=(0.299*R)+(0.587*G)+(0.114*B);
mat_yiq[i+1]=(0.596*R)-(0.275*G)-(0.321*B);
mat_yiq[i+2]=(0.212*R)-(0.523*G)+(0.311*B);
}
}
end=omp_get_wtime();
cout<<" time taken by 3 threads in Parallel is  "<<(end-start)<<'\n';
// Number of threads = 6
start=omp_get_wtime();
#pragma omp parallel num_threads(6)
{
#pragma omp for
for(int i=0;i<N*N*3;i+=3){
int R=mat[i];
int G=mat[i+1];
int B=mat[i+2];
mat_gray[i]=(R*0.21)+(G*0.72)+(B*0.07);
mat_yiq[i]=(0.299*R)+(0.587*G)+(0.114*B);
mat_yiq[i+1]=(0.596*R)-(0.275*G)-(0.321*B);
mat_yiq[i+2]=(0.212*R)-(0.523*G)+(0.311*B);
}
}
end=omp_get_wtime();
cout<<"time taken by 6 threads in Parallel is  "<<(end-start)<<'\n';
// Number of threads = 9
start=omp_get_wtime();
#pragma omp parallel num_threads(9)
{
#pragma omp for
for(int i=0;i<N*N*3;i+=3){
int R=mat[i];
int G=mat[i+1];
int B=mat[i+2];
mat_gray[i]=(R*0.21)+(G*0.72)+(B*0.07);
mat_yiq[i]=(0.299*R)+(0.587*G)+(0.114*B);
mat_yiq[i+1]=(0.596*R)-(0.275*G)-(0.321*B);
mat_yiq[i+2]=(0.212*R)-(0.523*G)+(0.311*B);
}
}
end=omp_get_wtime();
cout<<" time taken by 9 threads in Parallel is "<<(end-start)<<'\n';
// Number of threads = 12
```

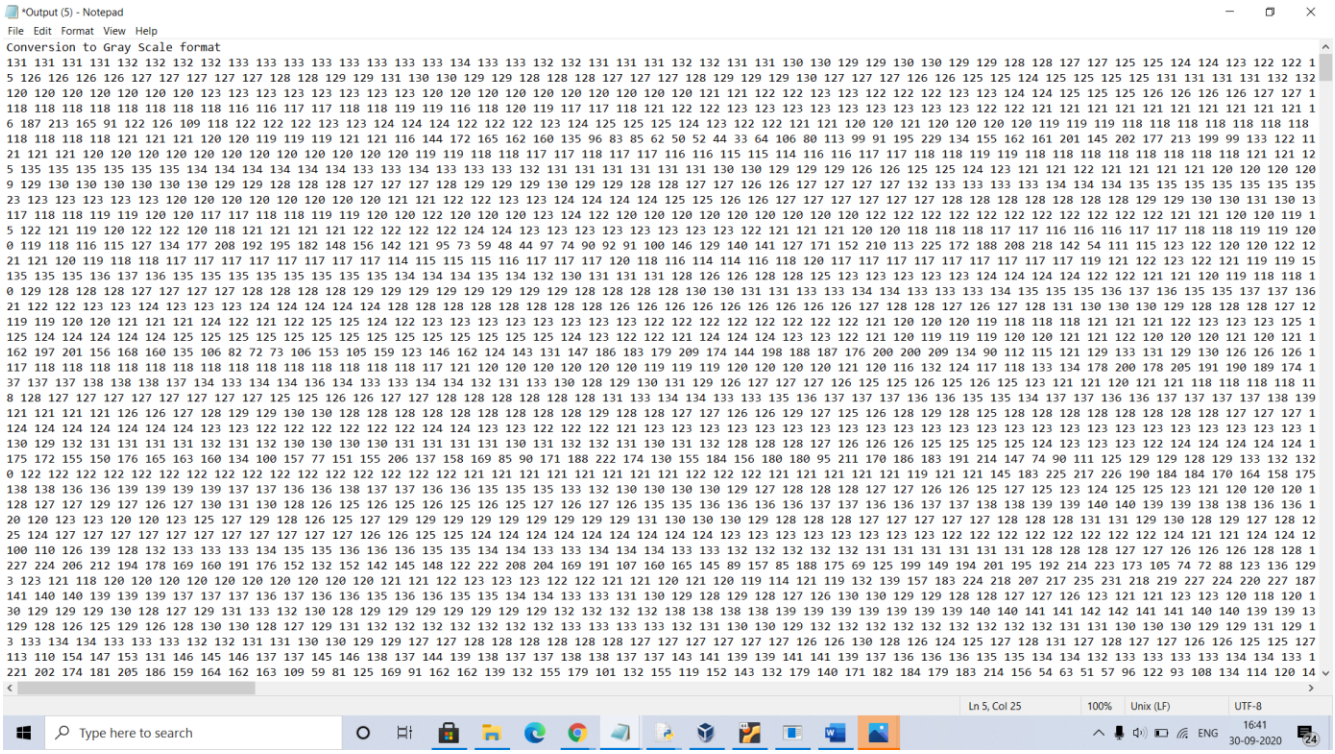
```
start=omp_get_wtime();
#pragma omp parallel num_threads(12)
{
#pragma omp for
for(int i=0;i<N*N*3;i+=3){
int R=mat[i];
int G=mat[i+1];
int B=mat[i+2];
mat_gray[i]=(R*0.21)+(G*0.72)+(B*0.07);
mat_yiq[i]=(0.299*R)+(0.587*G)+(0.114*B);
mat_yiq[i+1]=(0.596*R)-(0.275*G)-(0.321*B);
mat_yiq[i+2]=(0.212*R)-(0.523*G)+(0.311*B);
}
}
end=omp_get_wtime();
cout<<" time taken by 12 threads in Parallel is "<<(end-start)<<'\n';
freopen("Output.txt","w",stdout);
cout<<"Conversion to Gray Scale format\n";
for(int i=0;i<N*N;i+=3)
cout<<mat_gray[i]<<" ";
cout<<"\n\n\nConversion to YIQ format\n";
for(int i=0;i<N*N*3;i++)
cout<<mat_yiq[i]<<" ";
}
```

.

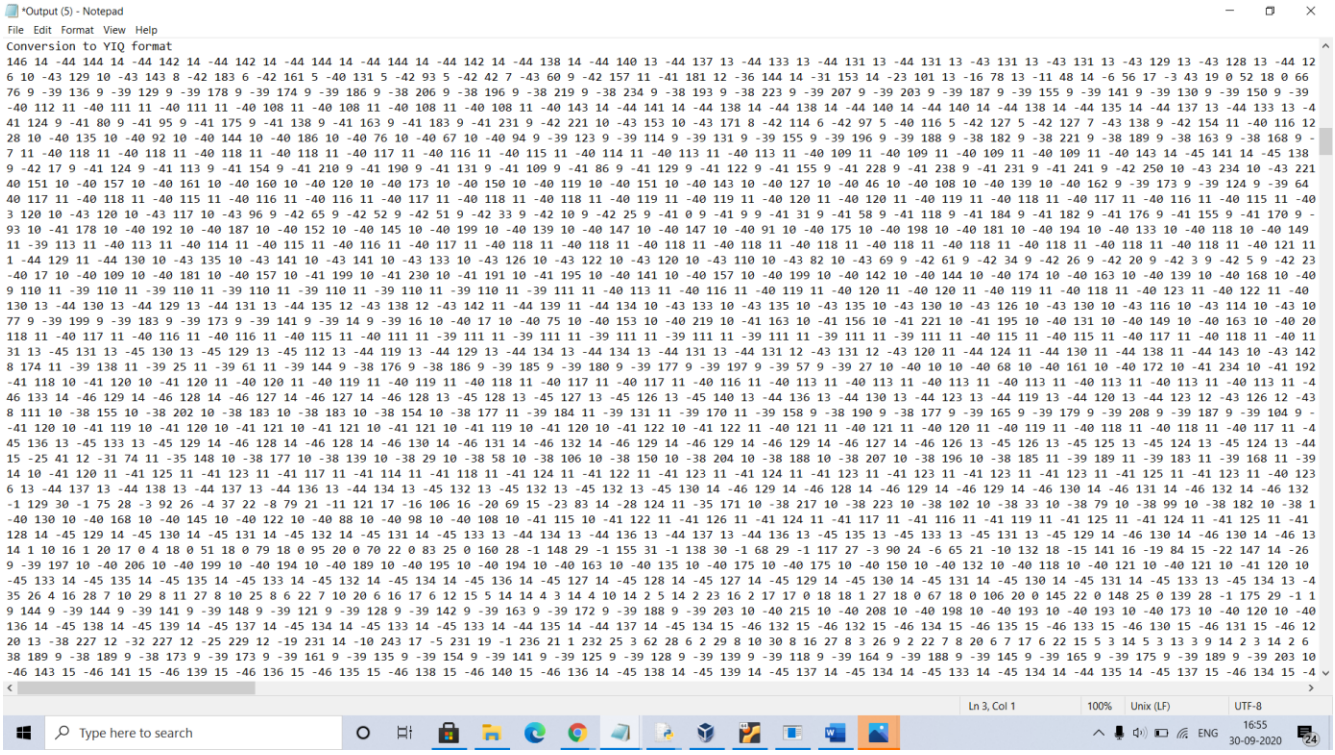


Output-:

There are over 300*300 iterations which can be time consuming for a single thread. In parallel execution time taken is lesser because the iterations are distributed among several threads(3, 6, 9 or 12). Hence, Parallel execution is taking lesser time than sequential



Gray scale conversion



YIQ scale conversion