Assignment 2(B)

Title: Write a program to implement Parallel Merge Sort using OpenMP

```
#include&lt;iostream&gt;
#include&lt;stdlib.h&gt;
#include&lt;omp.h&gt;
using namespace std;
void mergesort(int a[],int i,int j);
void merge(int a[],int i1,int j1,int i2,int j2);
void mergesort(int a[],int i,int j)
{
int mid;
if(i&lt;j)
{
mid=(i+j)/2;
#pragma omp parallel sections
{
#pragma omp section
{
mergesort(a,i,mid);
}
#pragma omp section
{
mergesort(a,mid+1,j);
}
}
merge(a,i,mid,mid+1,j);
}
}
void merge(int a[],int i1,int j1,int i2,int j2)
{
```

```
int temp[1000];


int i,j,k;
i=i1;
j=i2;
k=0;
while(i&amp;lt;=j1 &amp;amp;&amp;amp; j&amp;lt;=j2)
{
if(a[i]&amp;lt;a[j])
{
temp[k++]=a[i++];
}
else
{
temp[k++]=a[j++];
}
}
while(i&amp;lt;=j1)
{
temp[k++]=a[i++];
}
while(j&amp;lt;=j2)
{
temp[k++]=a[j++];
}
for(i=i1,j=0;i&amp;lt;=j2;i++,j++)
{
a[i]=temp[j];
}
}
```

```
int main()
{
int *a,n,i;
cout&lt;&lt;&quot;\n enter total no of elements=&gt;&quot;;
cin&gt;&gt;n;
a= new int[n];

cout&lt;&lt;&quot;\n enter elements=&gt;&quot;;
for(i=0;i&lt;n;i++)
{
cin&gt;&gt;a[i];
}
// start=.....
//#pragma omp…..
mergesort(a, 0, n-1);
// stop…….
cout&lt;&lt;&quot;\n sorted array is=&gt;&quot;;
for(i=0;i&lt;n;i++)
{
cout&lt;&lt;&quot;\n&quot;&lt;&lt;a[i];
}
// Cout&lt;&lt;Stop-Start
return 0;
}
```