```
/*
* ques2.c
* To sort a set of elements using Merge sort algorithm.
*   Created On: 4-19-2018
*       Author: shahique
*/

#include "stdafx.h"
#include<omp.h>
#include<time.h>
#include<stdio.h>
#include<iostream>
#include<conio.h>
#define max 100
void mergesort(int[100],int,int);
void merge(int[100],int,int,int);
int a[max];
void main()
{
    int i,n;
    clock_t s,e;
    printf("Enter the no.of elements\n");
    scanf_s("%d",&n);
    printf("Elements of the array before sorting\n");
    for(i=0;i<n;i++)
    {
        a[i]=rand()%1000;
        printf("%d\t",a[i]);
    }
    s=clock();
    mergesort(a,0,n1);
    e=clock();
    printf("\nElements of the array after sorting\n");
    #pragma omp parallel for
    for(i=0;i<n;i++)
        printf("%d\t",a[i]);
    printf("\nthe time taken=%f\n",(es)/CLK_TCK);
    getch();
}
void mergesort(int a[100],int low,int high)
{
    int mid;
    if(high>low)
    {
        mid=(low+high)/2;
        mergesort(a,low,mid);
        mergesort(a,mid+1,high);
        merge(a,low,mid,high);
    }
}
void merge(int a[100],int low,int mid,int high)
{
    int h=low,j=mid+1,i=low,b[max],k;
    #pragma omp parallel for
    while((h<=mid)&&(j<=high))
    {
        if(a[h]<=a[j])
        {
            b[i]=a[h];
            h=h+1;
        }
        else
        {
            b[i]=a[j];
            j=j+1;
        }
```

```
        i=i+1;
    }
    if(h>mid)
    {
        #pragma omp parallel for
        for(k=j;k<=high;k++)
        {
            b[i]=a[k];
            i++;
        }
    }
    else
    {
        #pragma omp parallel for
        for(k=h;k<=mid;k++)
        {
            b[i]=a[k];
            i++;
        }
    }
    #pragma omp parallel for
    for(k=low;k<=high;k++)
        a[k]=b[k];
}
```