**Mergesort.cpp:-->**

//mergesort using OPENMP

#include<iostream>

#include "omp.h"

using namespace std;

void merge(int a[],int l,int m,int r);

void mergesort(int a[],int l,int r);

int main()

{

int a[10],n,ctr;

cout<<"\nEnter the array size:\n";

cin>>n;

for(int i=0;i<n;i++)

{

cout<<"\nEnter the "<<i<<"th no:-\n";

cin>>a[i];

}

mergesort(a,0,n-1);

cout<<"\nThe sorted array is:--\n";

for(int i=0;i<n;i++)

{

cout<<a[i]<<" ";

}

cout<<"\nEXITING...\n";

return 0;

}

void merge(int a[],int l,int m,int r)

{

int i,j,k;

int n1=m-l+1;

int n2=r-m;

int L[n1],R[n2];

for(int i=0;i<n1;i++)

{

L[i]=a[l+i];

}

for(int j=0;j<n2;j++)

{

R[j]=a[m+1+j];

}

i=0;

j=0;

k=l;

while (i < n1 && j < n2)

{

if (L[i] <= R[j])

{

a[k] = L[i];

i++;

}

else

{

a[k] = R[j];

j++;

}

k++;

}

/\* Copy the remaining elements of L[], if there are any \*/

while (i < n1)

{

a[k] = L[i];

i++;

k++;

}

/\* Copy the remaining elements of R[], if there are any \*/

while (j < n2)

{

a[k] = R[j];

j++;

k++;

}

}

void mergesort(int a[],int l,int r)

{

if(l<r)

{

int m=l+(r-l)/2;

#pragma omp parallel sections

{

#pragma omp section

{

mergesort(a,l,m);

}

#pragma omp section

{

mergesort(a,m+1,r);

}

}

merge(a,l,m,r);

}

}

**OUTPUT:-->**

student@student-OptiPlex-390:~$ g++ -fopenmp mergesort.cpp

student@student-OptiPlex-390:~$ ./a.out

Enter the array size:

5

Enter the 0th no:-

45

Enter the 1th no:-

12

Enter the 2th no:-

65

Enter the 3th no:-

89

Enter the 4th no:-

22

The sorted array is:--

12 22 45 65 89

EXITING...

student@student-OptiPlex-390:~$