**Bubblesort.cpp:-->**

//bubble sort using open MP

#include<iostream>

#include "omp.h"

using namespace std;

void bubble(int \*a,int n);

void swap(int &e1,int &e2);

void bubble(int \*a,int n)

{

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

{

int f=i%2;

#pragma omp parallel for shared(a,f) num\_threads(5)

//write num\_threads(num) if required typically n-1 because of n-1 passes

for(int j=f;j<n-1;j=j+2)

{

if(a[j] > a[j+1])

{

swap(a[j],a[j+1]);

}

}

}

}

void swap(int &a,int &b)

{

int temp;

temp=a;

a=b;

b=temp;

}

int main()

{

int \*a;

int n,counter;

cout<<"\nEnter no of elements:- ";

cin>>n;

a=new int[n];

for(counter=0;counter<n;counter++)

{

cout<<"\nEnter "<<counter<<"th element:- ";

cin>>a[counter];

}

bubble(a,n);

cout<<"\nSorted array:-- \n";

for(counter=0;counter<n;counter++)

{

cout<<a[counter]<< " ";

}

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

return 0;

}

**OUTPUT:--**

student@student-OptiPlex-3020:~$ g++ -fopenmp bubblesort.cpp

student@student-OptiPlex-3020:~$ ./a.out

Enter no of elements:- 6

Enter 0th element:- 1

Enter 1th element:- 5

Enter 2th element:- 6

Enter 3th element:- 3

Enter 4th element:- 4

Enter 5th element:- 2

Sorted array:--

1 2 3 4 5 6

EXITING..

student@student-OptiPlex-3020:~$