**Merge Sort**

#include <iostream>

#include <omp.h>

using namespace std;

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

int i = l, j = m + 1, k = 0, temp[10000];

while (i <= m && j <= r) temp[k++] = (a[i] < a[j]) ? a[i++] : a[j++];

while (i <= m) temp[k++] = a[i++];

while (j <= r) temp[k++] = a[j++];

for (i = l, j = 0; i <= r; i++) a[i] = temp[j++];

}

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

if (l < r) {

int m = (l + r) / 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);

}

}

int main() {

int n, a[10000];

cout << "enter total no of elements:";

cin >> n;

cout << "enter elements:";

for (int i = 0; i < n; i++) cin >> a[i];

mergesort(a, 0, n - 1);

cout << "sorted array is:";

for (int i = 0; i < n; i++) cout << "\n" << a[i];

return 0;

}