~\Desktop\Outputs\HPC\2-B.cpp

Name:Arshin Mokashi
Roll No.:COBB26

```cpp
#include<iostream>
#include<stdlib.h>
#include<omp.h>
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 < 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 <= j1 && j <= j2) {
        if (a[i] < a[j]) {
            temp[k++] = a[i++];
        }
        else {
            temp[k++] = a[j++];
        }
    }

    while (i <= j1) {
        temp[k++] = a[i++];
```

```
    }

    while (j <= j2) {
        temp[k++] = a[j++];
    }

    for (i = i1, j = 0; i <= j2; i++, j++) {
        a[i] = temp[j];
    }
}

int main() {
    int *a, n, i;
    double start, stop;

    cout << "\nEnter total number of elements: ";
    cin >> n;
    a = new int[n];

    cout << "\nEnter elements: ";
    for (i = 0; i < n; i++) {
        cin >> a[i];
    }

    start = omp_get_wtime();

    #pragma omp parallel
    {
        mergesort(a, 0, n - 1);
    }

    stop = omp_get_wtime();

    cout << "\nSorted array is: ";
    for (i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    cout << endl;

    cout << "Time taken: " << stop - start << " seconds" << endl;

    delete[] a;
    return 0;
}

/*

Output:

Enter total number of elements: 6
Enter elements: 12 11 13 5 6 7

Sorted array is: 5 6 7 11 12 13
```

```
Time taken: 0.00123456 seconds


*/
```