## Multithreading Merge Sort in C++

```
#include <iostream>
#include <pthreaad.h>
using namespace std;

const int max_size = 200;
int array[max_size];
int size;
//function of merge sort
void merge_sort(int arr[], int Left, int mid, int Right)
{
int i, j, k;
int n1 = mid - Left + 1;
int n2 = Right - mid;
int Left_Array[n1], Right_Array[n2];

for (i = 0; i < n1; i++)
Left_Array[i] = arr[Left + i];

for (j = 0; j < n2; j++)
Right_Array[j] = arr[mid + 1 + j];

i = 0;
j = 0;
k = Left;

while (i < n1 && j < n2)

{
if (Left_Array[i] <= Right_Array[j])
{
arr[k] = Left_Array[i];
i++;
}

else
{
arr[k] = Right_Array[j];
j++;
}
k++;
}

while (i < n1)
{
arr[k] = Left_Array[i];
i++;
k++;
}

while (j < n2)
{
arr[k] = Right_Array[j];
j++;
k++;
}
}

void mergeSort(int arr[], int left, int right)
{
if (left < right)
{
int mid = left + (right - left) / 2;
mergeSort(arr, left, mid);
mergeSort(arr, mid + 1, right);
merge_sort(arr, left, mid, right);
}
}

//Function of multithreading mergesort
void *mergeSortThread(void *args)
{
int *arr = (int *) args;
int Left = arr[0];
int Right = arr[1];
mergeSort(array, Left, Right);
return 0;
}


int main()
{
cout << "Enter the size of  an array: "<<endl<<endl;
cin >> size;
cout<<endl;
cout << "Enter the elements of that array: "<<endl;
cout<<endl;
for (int i = 0; i < size; i++)
cin >> array[i];

pthread_t thread1, thread2;
int arr1[2] = {0, size/2 - 1};
int arr2[2] = {size/2, size - 1};

pthread_create(&thread1, NULL, mergeSortThread, (void *) arr1);
pthread_create(&thread2, NULL, mergeSortThread, (void *) arr2);
pthread_join(thread1, NULL);
pthread_join(thread2, NULL);
```

```
merge_sort(array, 0, size/2 - 1, size - 1);
cout <<"Sorted array is : "<<endl;
for (int i = 0; i < size; i++)
cout << array[i] << " ";
cout << endl;
return 0;
}
```