```cpp
#include <iostream>

using namespace std;


void merge(int array[], int left, int mid, int right)

{

        int subArrayOne = mid - left + 1;

        int subArrayTwo = right - mid;




        int *leftArray = new int[subArrayOne], *rightArray = new int[subArrayTwo];


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

                leftArray[i] = array[left + i];

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

                rightArray[j] = array[mid + 1 + j];


        int indexOfSubArrayOne = 0,

                indexOfSubArrayTwo = 0;

        int indexOfMergedArray = left;


        while (indexOfSubArrayOne < subArrayOne && indexOfSubArrayTwo < subArrayTwo) {

                if (leftArray[indexOfSubArrayOne] < rightArray[indexOfSubArrayTwo]) {

                        array[indexOfMergedArray] = leftArray[indexOfSubArrayOne];

                        indexOfSubArrayOne++;

                }

                else if (leftArray[indexOfSubArrayOne] > rightArray[indexOfSubArrayTwo]){
```

```
                array[indexOfMergedArray] = rightArray[indexOfSubArrayTwo];
                indexOfSubArrayTwo++;
        }
        else{
       array[indexOfMergedArray] = rightArray[indexOfSubArrayTwo];
                indexOfMergedArray++;
       array[indexOfMergedArray] = leftArray[indexOfSubArrayOne];

                indexOfSubArrayTwo++;
                indexOfSubArrayOne++;

        }
        indexOfMergedArray++;
}

while (indexOfSubArrayOne < subArrayOne) {
        array[indexOfMergedArray] = leftArray[indexOfSubArrayOne];
        indexOfSubArrayOne++;
        indexOfMergedArray++;
}

while (indexOfSubArrayTwo < subArrayTwo) {
        array[indexOfMergedArray] = rightArray[indexOfSubArrayTwo];
        indexOfSubArrayTwo++;
        indexOfMergedArray++;
}
```

```
}

void mergeSort(int array[], int    begin, int    end)
{
        if (begin >= end)
                return;

        int mid = begin + (end - begin) / 2;
        mergeSort(array, begin, mid);
        mergeSort(array, mid + 1, end);
        merge(array, begin, mid, end);
}

void printArray(int A[], int size)
{
        for (int i = 0; i < size; i++)
                cout << A[i] << " ";
}

int main()
{
        int arr[6] = { 12, 10, 12, 10, 10, 12 };


        cout << "Given array is \n";
        printArray(arr, 6);
```

```
    mergeSort(arr, 0, 5);


    cout << "\nSorted array is \n";

    printArray(arr, 6);

    return 0;

}
```

Output:

```
Given array is

12 10 12 10 10 12

Sorted array is

10 10 10 12 12 12
```

```
#include<iostream>

using namespace std;

#define SIZE 10

class Quick

{

	int arr[SIZE];

public:

	int get_data();

	void quicksort(int, int);

	int partition( int,int);

	void swap(int, int);

	void display(int);

};


int Quick::get_data()

{

	int i,n;

	cout<<"Enter total number of elements:";

	cin>>n;

	cout<<"Enter the percentage marks of each student:";

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

	{

		cin>>arr[i];

	}

	return n;

}
```

```
void Quick:: quicksort( int p,int q)

{

    int j;

    if(p<q)

    {

                    j=partition(p,q);

                    quicksort(p,j-1);

                    quicksort(j+1,q);


    }

}


int Quick:: partition(int start,int end_index)

{

          int low=start,high=end_index;

          int pivot =arr[start];

          do{

                while(arr[low]<=pivot)

                        low++;

                while(arr[high]>pivot)

                        high--;

                if(low<high){


                        swap(low,high);

                }
          }while(low<high);

          swap(start,high);



          return high;

}


void Quick:: swap(int i, int j)

{

        int temp;

        temp=arr[i];

        arr[i]=arr[j];

        arr[j]=temp;

}


  void Quick :: display(int n)

{

        cout<<"\n \t Percentage marks of top five students...\n";

        for(int i=n-1;i>=n-5;i--)

                cout<<" "<<arr[i];

}


int main()

{

        Quick obj;

        int n;
        cout<<"\n Quicksort Method \n";

        n=obj.get_data();


        obj.quicksort(0,n-1);

        obj.display(n);

        return 0;

}
```

Output:

```
Quicksort Method

Enter total number of elements:4

Enter the percentage marks of each student:67

89

67

87

Percentage marks of top five students...

  89 87 67 67
```