Heap Sort:

```
#include <iostream>

using namespace std;

void heapify(int A[],int i,int n)
{
    int largest = i;
    int left = 2*i + 1;
    int right = 2*i + 2;
    if(left < n && A[left] > A[largest])
        largest = left;
    if(right < n && A[right] > A[largest])
        largest = right;
    if(largest != i)
    {
        //swap A[i] and A[largest]
        swap(A[i],A[largest]);

        heapify(A,largest,n);
    }
}



void heapSort(int A[],int n)
{
    //buildHeap
    for(int i=n/2-1;i>=0;i--)
        heapify(A,i,n);

    for(int i=n-1;i>=0;i--)
    {
        //swap A[i] and A[0]
        int temp = A[i];
        A[i] = A[0];
        A[0] = temp;
        heapify(A,0,i);
    }

    cout << "Sorted." << endl;
    for(int i=0;i<10;i++)
    {
        cout << A[i] << " ";
```

```
    }
}
int main()
{
    int A[10]={0};
    cout << "Enter 10 elements: " << endl;
    for(int i=0;i<10;i++)
    {
        cin >> A[i];
    }

    heapSort(A,10);
}
```

Merge Sort:

```
#include <iostream>
void Combine(int arr[],int p,int q,int r)
{
    int i,j,k;
    int n1 = q-p+1;
    int n2 = r - q;
    int L[n1],R[n2];
    for(i=0; i < n1; i++)
    {
        L[i] = arr[p+i];
    }
    for(j=0; j < n2; j++)
    {
        R[j] = arr[q+j+1];
    }
    i=0; j=0;

    for(k = p;k <= r ; k++)
    {
        if(L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
    }
}


void MergeSort(int arr[],int p,int r)
{
    if(p<r)
    {
        int q = (p+r)/2;
        MergeSort(arr,p,q);
        MergeSort(arr,q+1,r);
        Combine(arr,p,q,r);
    }      }
```

```
int main()
{
    int arr[10];
    for(int i=0;i<10;i++)
    {
        cin >> arr[i];
    }
    MergeSort(arr,0,9);
    for(int i=0;i<10;i++)
        cout << arr[i] << " ";
}
```