

WEEK 5 ADA

VIBHA HUGAR

1BM21CS255

MERGE SORT CODE

```
#include<stdio.h>
```

```
void mergesort(int a[],int i,int j);
```

```
void merge(int a[],int i1,int j1,int i2,int j2);
```

```
int main()
```

```
{
```

```
int a[30],n,i;
```

```
printf("Enter no of elements:");
```

```
scanf("%d",&n);
```

```
printf("Enter array elements:");
```

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

```
scanf("%d",&a[i]);
```

```
mergesort(a,0,n-1);
```

```
printf("\nSorted array is :");
```

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

```
printf("%d ",a[i]);
```

```
return 0;
```

```
}
```

```
void mergesort(int a[],int i,int j)
```

```
{
```

```
int mid;
```

```

if(i<j)
{
mid=(i+j)/2;
mergesort(a,i,mid); //left recursion
mergesort(a,mid+1,j); //right recursion
merge(a,i,mid,mid+1,j); //merging of two sorted sub-arrays
}
}

```

```

void merge(int a[],int i1,int j1,int i2,int j2)
{
int temp[50]; //array used for merging
int i,j,k;
i=i1; //beginning of the first list
j=i2; //beginning of the second list
k=0;
while(i<=j1 && j<=j2) //while elements in both lists
{
if(a[i]<a[j])
temp[k++]=a[i++];
else
temp[k++]=a[j++];
}
while(i<=j1) //copy remaining elements of the first list
temp[k++]=a[i++];
while(j<=j2) //copy remaining elements of the second list
temp[k++]=a[j++];
//Transfer elements from temp[] back to a[]
for(i=i1,j=0;i<=j2;i++,j++)

```

```
a[i]=temp[j];  
}
```

MERGE-SORT OUTPUT

 "C:\Users\Admin\Desktop\cs255\4th sem ada lab\mergesort.exe"

```
Enter no of elements:5  
Enter array elements:3 4746 5 865 2  
  
Sorted array is :2 3 5 865 4746  
Process returned 0 (0x0)   execution time : 9.671 s  
Press any key to continue.
```

 "C:\Users\Admin\Desktop\cs255\4th sem ada lab\mergesort.exe"

```
Enter no of elements:3  
Enter array elements:45 5 7  
  
Sorted array is :5 7 45  
Process returned 0 (0x0)   execution time : 4.647 s  
Press any key to continue.
```

 "C:\Users\Admin\Desktop\cs255\4th sem ada lab\mergesort.exe"

```
Enter no of elements:8  
Enter array elements:3 4 656 75 34234 6 455 23  
  
Sorted array is :3 4 6 23 75 455 656 34234  
Process returned 0 (0x0)   execution time : 30.783 s  
Press any key to continue.
```

QUICKSORT CODE

```
#include<stdio.h>
```

```
void qsort(int a[], int low, int high)
```

```
{  
    int mid;  
    if(low<high)  
    {  
        mid=partition(a,low,high);  
        qsort(a,low,mid-1);  
        qsort(a,mid+1, high);  
    }  
}
```

```
int partition(int a[],int low, int high)
```

```
{  
    int i,j,temp, pivot;  
    pivot=a[low];  
    i=low+1;  
    j=high;  
  
    while(i<=j)  
    {  
        while(a[i]<=pivot)  
            i++;  
        while(a[j]>pivot)  
            j--;  
  
        if(i<j)
```

```
{  
    temp=a[i];  
    a[i]=a[j];  
    a[j]=temp;  
}  
}
```

```
temp=a[low];  
a[low]=a[j];  
a[j]=temp;  
return j;  
}
```

```
int main()  
{  
    int a[30],n,i;  
    printf("Enter no of elements:");  
    scanf("%d",&n);  
    printf("Enter array elements:");  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
  
    qsort(a,0,n-1);  
    printf("\nSorted array is :");  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
  
    return 0;  
}
```

QUICK-SORT OUTPUT

 "C:\Users\Admin\Desktop\cs255\4th sem ada lab\quicksort.exe"

```
Enter no of elements:5
```

```
Enter array elements:4 5675 67 4 6
```

```
Sorted array is :4 4 6 67 5675
```

```
Process returned 0 (0x0)   execution time : 7.235 s
```

```
Press any key to continue.
```