

Name:Gaurav singh

Roll No: 70

Semester: 3 Sem.

Subject: Data structures and Algorithms.

Practical: 8 Develop the programs for the following sorting algorithms:

1. Merge Sort
2. Quick Sort

1. MERGE SORT:

```
#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);
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[50];
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];
}

```

Output Screen:

The screenshot shows a C++ IDE with a file named 'main.c'. The code implements a merge sort algorithm. The 'Output' window on the right shows the program's execution: it prompts for the number of elements (5) and array elements (60), then displays the sorted array: 1 9 32 56 60.

```

main.c
33 void merge(int a[],int i1,int j1,int i2,int j2)
34 {
35     int temp[50];
36     int i,j,k;
37     i=i1;
38     j=j1;
39     k=0;
40     while(i<=j1 && j<=j2)
41     {
42         if(a[i]<a[j])
43             temp[k++]=a[i++];
44         else
45             temp[k++]=a[j++];
46     }
47     while(i<=j1)
48         temp[k++]=a[i++];
49     while(j<=j2)
50         temp[k++]=a[j++];
51     for(i=i1,j=0;i<=j2;i++,j++)
52         a[i]=temp[j];
53 }

```

Output

```

/tmp/xSdiu1rz0E.o
Enter no of elements:5
Enter array elements:60
1
32
56
9
Sorted array is :1 9 32 56 60

```

Activate Windows
Go to Settings to activate Windows.

2. QUICK SORT:

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

```
void quick_sort(int[],int,int);
```

```
int partition(int[],int,int);
```

```
int main()
```

```
{
```

```
int a[50],n,i;
```

```
printf("How many elements?");
```

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

```
printf("\nEnter array elements:");
```

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

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

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

```
printf("\nArray after sorting:");
```

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

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

```
return 0;
```

```
}
```

```
void quick_sort(int a[],int l,int u)
```

```
{
```

```
int j;
```

```
if(l<u)
```

```
{
```

```
j=partition(a,l,u);
```

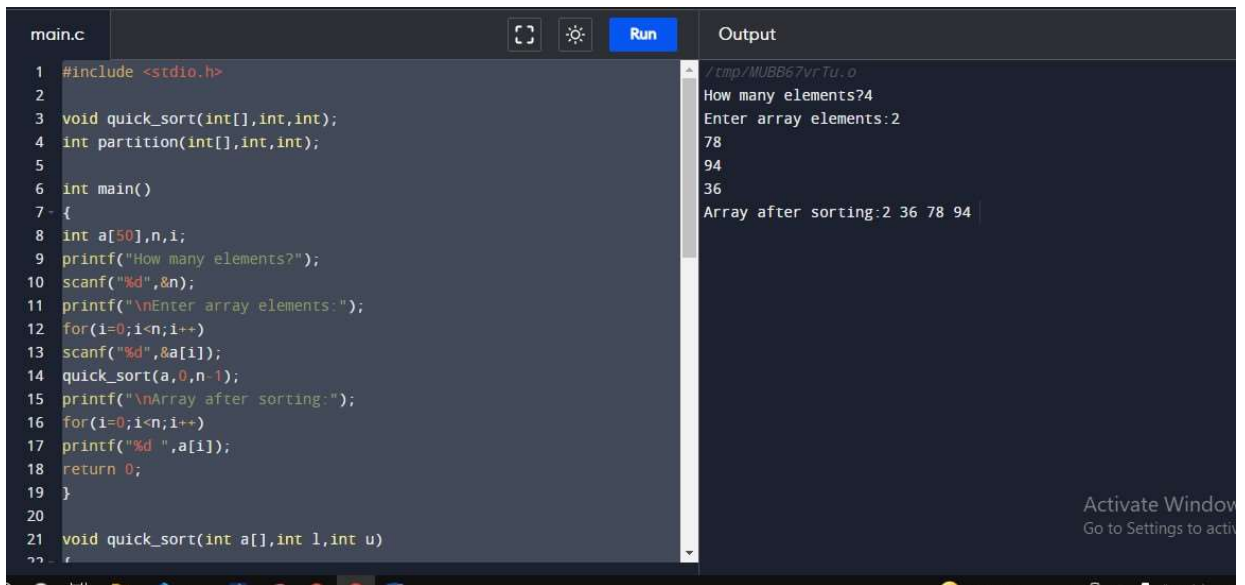
```
quick_sort(a,l,j-1);
```

```
quick_sort(a,j+1,u);  
}  
}
```

```
int partition(int a[],int l,int u)  
{  
    int v,i,j,temp;  
    v=a[l];  
    i=l;  
    j=u+1;  
    do  
    {  
        do  
        i++;  
        while(a[i]<v&&i<=u);  
        do  
        j--;  
        while(v<a[j]);  
        if(i<j)  
        {  
            temp=a[i];  
            a[i]=a[j];  
            a[j]=temp;  
        }  
    }while(i<j);  
    a[l]=a[j];  
    a[j]=v;  
    return(j);  
}
```

}

Output Screen:



The screenshot shows a C++ IDE with a file named 'main.c'. The code implements a quick sort algorithm. The 'Output' panel on the right shows the program's execution: it prompts for the number of elements (4), reads the array elements (78, 94, 36), and displays the sorted array (2, 36, 78, 94). The code in 'main.c' includes `<stdio.h>`, defines `quick_sort` and `partition` functions, and uses `scanf` and `printf` for input and output. The `quick_sort` function is recursive, and the `partition` function is used to divide the array into two parts. The `main` function calls `quick_sort` and prints the sorted array.

```
main.c
1 #include <stdio.h>
2
3 void quick_sort(int[],int,int);
4 int partition(int[],int,int);
5
6 int main()
7 {
8     int a[50],n,i;
9     printf("How many elements?");
10    scanf("%d",&n);
11    printf("\nEnter array elements:");
12    for(i=0;i<n;i++)
13        scanf("%d",&a[i]);
14    quick_sort(a,0,n-1);
15    printf("\nArray after sorting:");
16    for(i=0;i<n;i++)
17        printf("%d ",a[i]);
18    return 0;
19 }
20
21 void quick_sort(int a[],int l,int u)
22 {
```

Output

```
// cmp/MUBB67vrTu.o
How many elements?4
Enter array elements:2
78
94
36
Array after sorting:2 36 78 94
```

Activate Windows
Go to Settings to activate Windows.