

DATA STRUCTURES-CSA0396

19. Write a C program to arrange a series of numbers using Quick Sort

CODING:

```
#include<stdio.h>

void quicksort(int [],int,int);

int main()
{
    int a[20],n,i;
    printf("Enter size of the array: ");
    scanf("%d",&n);
    printf("Enter %d elements: ",n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    quicksort(a,0,n-1);
    printf("Sorted elements: ");
    for(i=0;i<n;i++)
        printf(" %d",a[i]);
    return 0;
}

void quicksort(int a[10],int first,int last)
{
    int pivot,j,t,i;
```

```
    if(first<last)
    {
        pivot=first;
        i=first;
        j=last;

        while(i<j)
        {
            while(a[i] <= a[pivot])
                i++;

            while(a[j]>a[pivot])
                j--;

            if(i<j)
            {
                t=a[i];
                a[i]=a[j];
                a[j]=t;
            }
        }
    }
```

```

t=a[pivot];

    a[pivot]=a[j];

    a[j]=t;


    quicksort(a,first,j-1);

    quicksort(a,j+1,last);

}

}

```

OUTPUT:

The screenshot shows a C++ IDE with a file named 'main.cpp'. The code implements a quicksort algorithm. The main function prompts the user to enter the size of the array (10) and then 10 elements (87, 98, 41, 20, 4, 65, 45, 21, 90, 100). It then prints the sorted elements: 4 5 20 21 41 65 87 90 98 100.

```

main.cpp
1  #include<stdio.h>
2  void quicksort(int [],int,int);
3  int main()
4  {
5      int a[20],n,i;
6      printf("Enter size of the array: ");
7      scanf("%d",&n);
8      printf("Enter %d elements: ",n);
9      for(i=0;i<n;i++)
10         scanf("%d",&a[i]);
11     quicksort(a,0,n-1);
12     printf("Sorted elements: ");
13     for(i=0;i<n;i++)
14         printf(" %d",a[i]);
15     return 0;
16 }
17 void quicksort(int a[10],int first,int last)
18 {
19     int pivot,j,t,i;
20
21
22     if(first<last)
23     {
24         pivot=first;
25         i=first;
26         j=last;

```

Output

```

/tmp/tNddf8610S.o
Enter size of the array: 10
Enter 10 elements: 87
98
41
20
4
65
45
21
90
100
Sorted elements:  4 5 20 21 41 65 87 90 98 100

```