DATA STRUCTURES-CSA0396

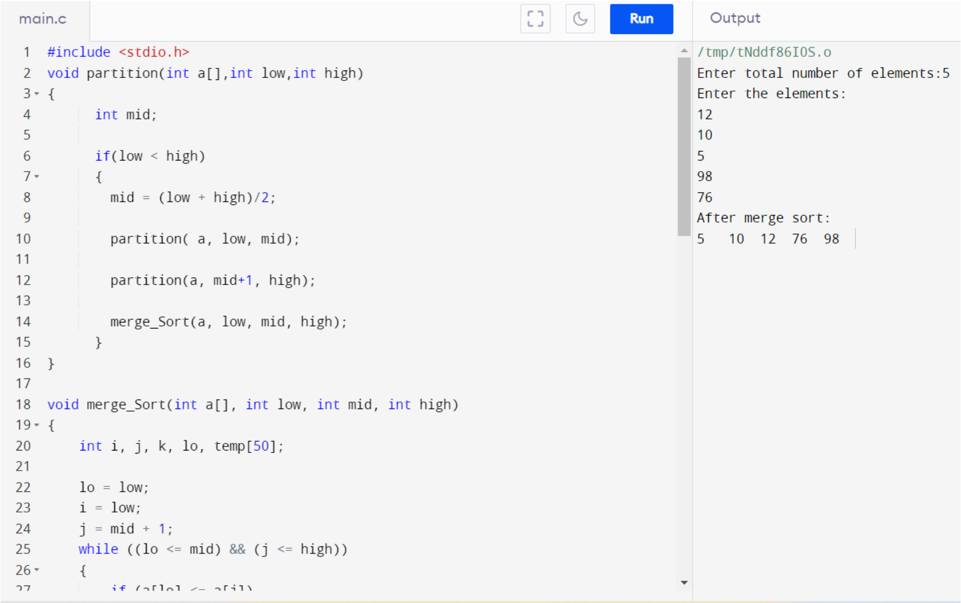
18. Write a C program to arrange a series of numbers using Merge Sort

CODING:

```
#include <stdio.h>
void partition(int a[],int low,int high)
{
    int mid;

  if(low < high)
    {
mid = (low + high)/2;

partition( a, low, mid);

partition(a, mid+1, high);

merge_Sort(a, low, mid, high);
  }
}

void merge_Sort(int a[], int low, int mid, int high)
{
int i, j, k, lo, temp[50];

lo = low;
i = low;
```

```
j = mid + 1;

while ((lo <= mid) && (j <= high))

{

if (a[lo] <= a[j])

{

temp[i] = a[lo];

lo++;

}

else

{

temp[i] = a[j];

j++;

}

i++;

}


if (lo > mid)

{

for (k = j; k <= high; k++)

{

temp[i] = a[k];

i++;

}

}

else

{

for (k = lo; k <= mid; k++)

{

temp[i] = a[k];
i++;
}
}

for (k = low; k <= high; k++)
a[k] = temp[k];
}



int main()
{
    int a[50] ,  i, n;
    printf("Enter total number of elements:");
    scanf("%d", &n);
    printf("Enter the elements:\n");
    for(i = 0; i < n; i++)
        scanf("%d", &a[i]);

    partition( a, 0, n - 1);

    printf("After merge sort:\n");
    for(i = 0;i < n; i++)
                printf("%d\t", a[i]);
}
```

OUTPUT:



```c
#include <stdio.h>
void partition(int a[],int low,int high)
{
      int mid;

      if(low < high)
      {
        mid = (low + high)/2;

        partition( a, low, mid);

        partition(a, mid+1, high);

        merge_Sort(a, low, mid, high);
      }
}

void merge_Sort(int a[], int low, int mid, int high)
{
    int i, j, k, lo, temp[50];

    lo = low;
    i = low;
    j = mid + 1;
    while ((lo <= mid) && (j <= high))
    {
        if (a[lo] <= a[j])
```

```
/tmp/tNddf86IOS.o
Enter total number of elements:5
Enter the elements:
12
10
5
98
76
After merge sort:
5   10  12  76  98
```