WEEK 4

Sort a given set of N integer elements using Merge Sort technique

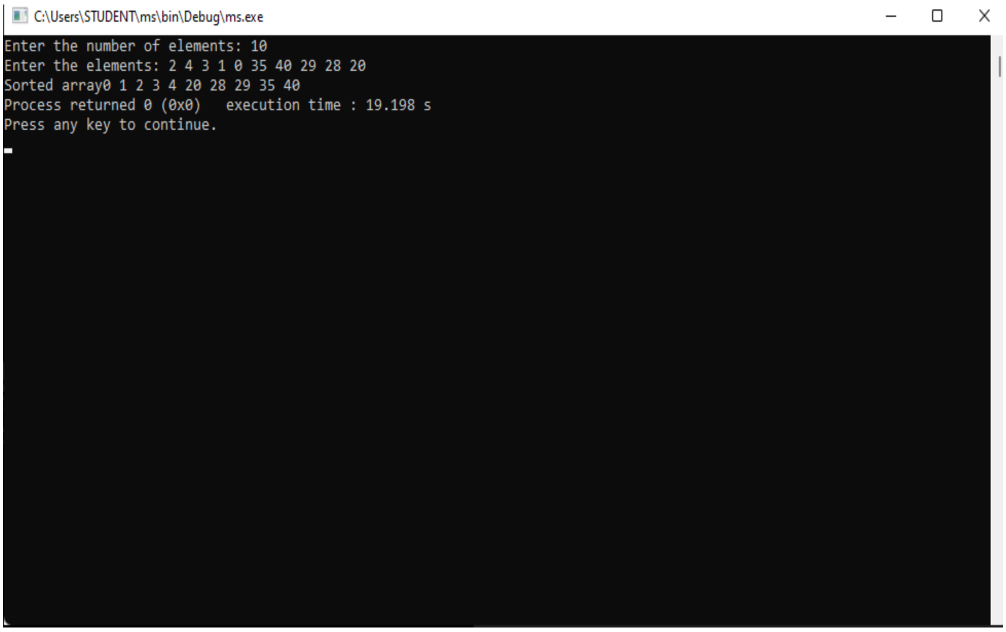
CODE:

```
#include <stdio.h>
#include <stdlib.h>
void merge(int low,int mid,int high,int a[20],int m[20])
{
   int i = low;
   int j = mid+1;
   int k = 0;
   while(i<=mid && j<=high)
   {
      if(a[i]<a[j])
      {
         m[k] = a[i];
         i++;
         k++;
      }
      else
      {
         m[k] = a[j];
         j++;
         k++;
      }
```

```
    }
    while (i <= mid)
    {
        m[k] = a[i];
        i++;
        k++;
    }
    while (j <= high)
    {
        m[k] = a[j];
        j++;
        k++;
    }
    for(int i=0;i<k;i++)
        a[low+i] = m[i];
}
void merge_sort(int low,int high,int a[20],int merged[20])
{
    if(low<high)
    {
        int mid = (low+high)/2;
        merge_sort(low,mid,a,merged);
        merge_sort(mid+1,high,a,merged);
        merge(low,mid,high,a,merged);
    }
}
```

```
int main()
{
   int n,a[30];
   printf("Enter the number of elements:");
   scanf("%d",&n);
   printf("Enter the elements:");
   for(int i=0;i<n;i++)
     scanf("%d",&a[i]);
   int merged[30];
   merge_sort(0,n-1,a,merged);
   printf("Sorted array");
   for(int i=0;i<n;i++)
     printf("%d ",a[i]);
}
```

OUTPUT :



```
Enter the number of elements: 10
Enter the elements: 2 4 3 1 0 35 40 29 28 20
Sorted array0 1 2 3 4 20 28 29 35 40
Process returned 0 (0x0)   execution time : 19.198 s
Press any key to continue.
```