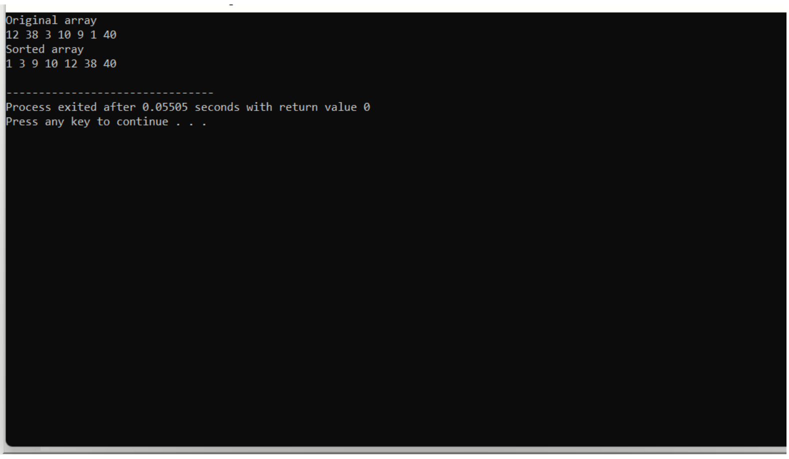
```
#include <stdio.h>
void merge(int arr[], int start, int mid, int end)
 {
int i,j,k;
 int len1 = mid - start + 1;
 int len2 = end - mid;

 int leftArr[len1], rightArr[len2];

 for ( i = 0; i < len1; i++)
  leftArr[i] = arr[start + i];
 for (j = 0; j < len2; j++)
  rightArr[j] = arr[mid + 1 + j];
 i = 0;
 j = 0;
 k = start;

 while (i < len1 && j < len2)
 {
  if (leftArr[i] <= rightArr[j])
       {
   arr[k] = leftArr[i];
   i++;
  }
       else
       {
   arr[k] = rightArr[j];
   j++;
  }
  k++;
 }
```

```
  while (i < len1) {
    arr[k] = leftArr[i];
    i++;
    k++;
  }

  while (j < len2) {
    arr[k] = rightArr[j];
    j++;
    k++;
  }
}

void mergeSort(int arr[], int start, int end) {
  if (start < end) {

    int mid = start + (end - start) / 2;
    mergeSort(arr, start, mid);
    mergeSort(arr, mid + 1, end);
    merge(arr, start, mid, end);
  }
}

void display(int arr[], int size)
 {
  int i;
  for (i = 0; i < size; i++)
  printf("%d ", arr[i]);
  printf("\n");
}

int main() {
  int arr[] = {12, 38, 3, 10, 9, 1 , 40};

  int size = sizeof(arr) / sizeof(arr[0]);


  printf("Original array\n");

  display(arr, size);


  mergeSort(arr, 0, size - 1);


  printf("Sorted array\n");

  display(arr, size);


}
```



```
Original array
12 38 3 10 9 1 40
Sorted array
1 3 9 10 12 38 40

--------------------------------
Process exited after 0.05505 seconds with return value 0
Press any key to continue . . .
```