

Program

```
#include <stdio.h>

void merge(int arr[], int l, int m, int r) {
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    i = 0;
    j = 0;
    k = l;

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        } else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int l, int r) {
    if (l < r) {
        int m = l + (r - l) / 2;

        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}

int main() {
    int arr[50], n, i;

    printf("Enter size of the array: ");
    scanf("%d", &n);

    printf("Enter %d elements: ", n);
```

MERGE SORT

Aim:

To sort a given array using merge sort.

Algorithm:

1. Start
2. create a function merge

```
n1 = mid - left + 1
n2 = right - mid
```

3. Create temporary arrays left half[n1], right half[n2]
4. Copy data to temporary arrays left half and right half

```
for i = 0 to n1 - 1:
    leftHalf[i] = arr[left + i]
for j = 0 to n2 - 1:
    rightHalf[j] = arr[mid + 1 + j]
```

5. Merge the temporary arrays back into arr[left..right]

```
i = 0, j = 0, k = left
while i < n1 and j < n2:
    if leftHalf[i] <= rightHalf[j]:
        arr[k] = leftHalf[i]
        i++
    else:
        arr[k] = rightHalf[j]
        j++
    k++
```

6. Copy the remaining elements of leftHalf, if any

```
arr[k] = left half[i]
i++
k++
```

```
    for (i = 0; i < n; i++) {  
        scanf("%d", &arr[i]);  
    }  
  
    mergeSort(arr, 0, n - 1);  
  
    printf("\nSorted array: ");  
    for (i = 0; i < n; i++) {  
        printf("%d ", arr[i]);  
    }  
  
    return 0;  
}
```

Output

```
Enter size of the array:5  
Enter 5 elements: 21 46 78 31 65  
Sorted array:21 31 46 65 78
```

7. Copy the remaining elements of `rightHalf`, if any

```
while j < n2:  
    arr[k] = right_half[j]  
    j++, k++
```

8. create function mergesort

```
if left < right:  
    mid = (left + right) / 2  
    mergesort(arr, left, mid)  
    mergesort(arr, mid + 1, right)  
    merge(arr, left, mid, right)
```

9. Stop

Result:

Program has been executed successfully and obtained the output