**// 1. MERGE SORT IN C**

#include <stdio.h>

// Function to merge two halves of an array

void merge(int arr[], int low, int mid, int high)

{

int b[10]; // Temporary array

int i = low, j = mid + 1, k = 0;

// Merge elements in sorted order

while (i <= mid && j <= high) {

if (arr[i] < arr[j])

{

b[k] = arr[i];

i++;

}

else

{

b[k] = arr[j];

j++;

}

k++;

}

// Copy remaining elements from left subarray

while (i <= mid)

{

b[k] = arr[i];

i++;

k++;

}

// Copy remaining elements from right subarray

while (j <= high)

{

b[k] = arr[j];

j++;

k++;

}

// Copy merged elements back to original array

for (i = low, k = 0; i <= high; i++, k++)

arr[i] = b[k];

}

// Function to perform merge sort

void mergeSort(int arr[], int left, int right)

{

if (left < right)

{

int mid = (left + right) / 2; // Find the middle point

// Recursively sort first and second halves

mergeSort(arr, left, mid);

mergeSort(arr, mid + 1, right);

// Merge the sorted halves

merge(arr, left, mid, right);

}

}

// Main function

int main()

{

int a[25],i,n;

printf("enter n value");

scanf("%d",&n);

printf("enter %d elements\n",n);

for(i=0;i<n;i++)

scanf("%d",&a[i]);

mergeSort(a, 0, n - 1);

printf("Sorted array:\n");

for(i=0;i<n;i++)

printf("%d\t",a[i]);

return 0;

}

**// 2.RADIX SORT IN C**

#include<stdio.h>

void radix(int[],int);

void radix(int a[],int n)

{

int max,digits=0,i,j,k,count[10],bucket[10][n],div=1,pos,steps;

max=a[0];

for(i=0;i<n;i++)

{

if(a[i]>max)

{

max=a[i];

}

}

while(max>0)

{

digits ++; max=max/10;

}

for(steps=1;steps<=digits;steps++)

{

for(i=0;i<10;i++)

{

count[i]=0;

}

for(i=0;i<n;i++)

{

pos=(a[i]/div)%10;

bucket[pos][count[pos]++]=a[i];

} k=0;

for(i=0;i<10;i++)

{

for(j=0;j<count[i];j++)

{

a[k]=bucket[i][j];

k++;

}

} div=div\*10;

}

}

int main()

{

int arr[50],n,i;

printf("enter n:");

scanf("%d",&n);

printf("enter %d values:",n);

for(i=0;i<n;i++)

{

scanf("%d",&arr[i]);

}

radix(arr,n);

printf("sorted array: ");

for(i=0;i<n;i++)

{

printf("%d ",arr[i]);

}printf("\n ");

return 0;

}