

Aim:

Write a program to **sort** (**ascending order**) the given elements using **radix sort** technique.

At the time of execution, the program should print the message on the console as:

Enter array size :

For example, if the user gives the **input** as:

Enter array size : 5

Next, the program should print the following message on the console as:

Enter 5 elements :

if the user gives the **input** as:

Enter 5 elements : 34 67 12 45 22

then the program should **print** the result as:

Before sorting the elements are : 34 67 12 45 22

After sorting the elements are : 12 22 34 45 67

Note: Do use the **printf()** function with a **newline** character (**\n**).

Source Code:**RadixSortMain2.c**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int size;
    int *arr,i;
    printf("Enter array size : ");
    scanf("%d",&size);
    arr = (int*) malloc(size * sizeof(int));
    printf("Enter %d elements : ",size);
    for(i=0;i<size;i++)
    {
        scanf("%d",&arr[i]);
    }
    printf("Before sorting the elements are : ");
    printArray(arr,size);
    RadixSort(arr,size);
    printf("After sorting the elements are : ");
    printArray(arr,size);
}
int largest(int a[], int n)
{
    int i,k=a[0];
    for(i=1;i<n;i++)
```

```
{
    if(a[i]>k)
    {
        k=a[i];
    }
}
return k;
}
void printArray(int a[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d ",a[i]);
    }
    printf("\n");
}
void RadixSort(int a[], int n)
{
    int bucket[10][10],bucket_count[10],i,j,k,rem,NOP=0,divi=1,large,pass;
    large=largest(a,n);
    while(large>0)
    {
        NOP++;
        large/=10;
    }
    for(pass=0;pass<NOP;pass++)
    {
        for(i=0;i<=10;i++)
        {
            bucket_count[i]=0;
        }
        for(i=0;i<n;i++)
        {
            rem=(a[i]/divi)%10;
            bucket[rem][bucket_count[rem]]=a[i];
            bucket_count[rem]++;
        }
        i=0;
        for(k=0;k<10;k++)
        {
            for(j=0;j<bucket_count[k];j++)
            {
                a[i]=bucket[k][j];
                i++;
            }
        }
        divi*=10;
    }
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output
Enter array size : 5
Enter 5 elements : 23
43
54
12
65
Before sorting the elements are : 23 43 54 12 65
After sorting the elements are : 12 23 43 54 65

Test Case - 2
User Output
Enter array size : 7
Enter 7 elements : 23
54
136
85
24
65
76
Before sorting the elements are : 23 54 136 85 24 65 76
After sorting the elements are : 23 24 54 65 76 85 136