RADIX SORT

Exp. No.:
AIM:

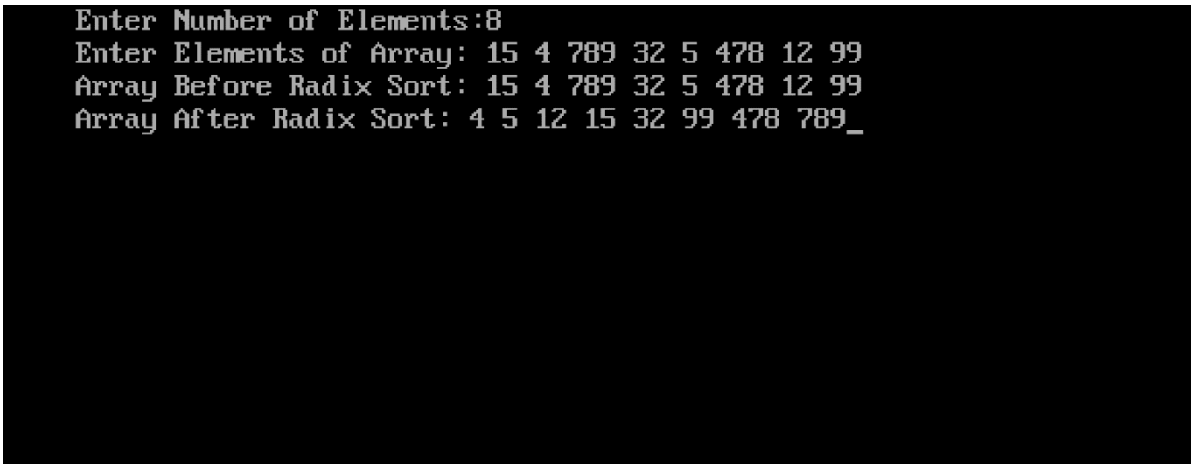
ALGORITHM:

PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

void countSort(int *a, int n, int P);

void radixSort(int *a, int n);

int main()
{
   int *a, i, n;
   printf("Enter Number of Elements:");
   scanf("%d", &n);
   a = (int *)malloc(n * sizeof(int));
   printf("Enter Elements of Array: ");
   for (i = 0; i < n; i++)
   {
      scanf("%d", &a[i]);
   }
   printf("Array Before Radix Sort:");
   for (i = 0; i < n; i++)
   {
      printf(" %d", a[i]);
   }
   radixSort(a, n);
   printf("\nArray After Radix Sort:");
   for (i = 0; i < n; i++)
   {
      printf(" %d", a[i]);
   }
   getch();
   clrscr();
   return 0;
}

void countSort(int *a, int n, int P)
```

```
{
    int i, *b, c[10] = {0};
    b = (int *)calloc(n, sizeof(int));
    for (i = 0; i < n; i++)
    {
        c[(a[i] / P) % 10]++;
    }
    for (i = 1; i < 10; i++)
    {
        c[i] += c[i - 1];
    }
    for (i = n - 1; i >= 0; i--)
    {
        b[c[(a[i] / P) % 10] - 1] = a[i];
        c[(a[i] / P) % 10]--;
    }
    for (i = 0; i < n; i++)
    {
        a[i] = b[i];
    }
}

void radixSort(int *a, int n)
{
    int P, i, max = a[0];
    for (i = 1; i < n; i++)
        if (max < a[i])
            max = a[i];
    for (P = 1; max / P > 0; P *= 10)
    {
        countSort(a, n, P);
    }
}
```

OUTPUT:



RESULT: