Practical 4

(Tanu Soni,88067)

Implement Radix Sort.

Code

```
#include <iostream>
using namespace std;

void printArray(int arr[], int n){
 cout<<"Sorted Array is : "<< " ";
 for(int i=0;i<n;i++){
    cout<<arr[i]<< " ";
 }
}
///function to get maximum value in arr[]
int getMax(int arr[], int n)
{
      int max = arr[0];
      for (int i = 1; i < n; i++)
            if (arr[i] > max)
                  max = arr[i];
      return max;
}

///counting sort
void countSort(int arr[], int n, int exp)
{
      int b[n]; /// output array
      int i, c[10] = {0};

      /// Store count of occurrences in count[]
      for (i = 0; i < n; i++)
            c[(arr[i] / exp) % 10]++;

      /// Change c[i] so that c[i] now contains actual position of this digit in output[]
      for (i = 1; i < 10; i++)
            c[i] += c[i - 1];
```

```
        /// Build the output array
        for (i = n - 1; i >= 0; i--) {
                b[c[(arr[i] / exp) % 10] - 1] = arr[i];
                c[(arr[i] / exp) % 10]--;
        }

        /// Copy the output array to arr[], so that arr[] now contains sorted
numbers according to current digit
        for (i = 0; i < n; i++)
                arr[i] = b[i];
}

/// Radix Sort
void radixSort(int arr[], int n)
{
        /// Find the maximum number to know number of digits
        int m = getMax(arr, n);
/*
         Do counting sort for every digit. Note that instead
         of passing digit number, exp is passed. exp is 10^i
         where i is current digit number
*/
        for (int exp = 1; m / exp > 0; exp *= 10)
                countSort(arr, n, exp);
}

int main()
{
        int arr[] = { 170, 45, 75, 90, 802, 24, 2, 66 };
        int n = 8;
        radixSort(arr, n);
        printArray(arr, n);
        return 0;
}
```

Output

```
Sorted Array is :  2 24 45 66 75 90 170 802
```