LAB-3

1)WAP to implement count sort

```
//count sort
 #include <iostream>
using namespace std;

void countSort(int inputArray[], int N)
{
    int M = 0;
   for (int i = 0; i < N; i++)
      M = max(M, inputArray[i]);

   int* countArray = new int[M + 1]();

   for (int i = 0; i < N; i++)
      countArray[inputArray[i]]++;

   for (int i = 1; i <= M; i++)
      countArray[i] += countArray[i - 1];
   int* outputArray = new int[N];

   for (int i = N - 1; i >= 0; i--)
   {
      outputArray[countArray[inputArray[i]] - 1] = inputArray[i];
      countArray[inputArray[i]]--;
   }
```

```
    for (int i = 0; i < N; i++)
        inputArray[i] = outputArray[i];
    delete[] countArray;
    delete[] outputArray;
}

int main()
{
    int N;
    cout << "Enter the number of elements: ";
    cin >> N;

    int* inputArray = new int[N];

    cout << "Enter the elements: ";
    for (int i = 0; i < N; i++)
        cin >> inputArray[i];

    countSort(inputArray, N);
    cout << "Sorted array: ";
    for (int i = 0; i < N; i++)
        cout << inputArray[i] << " ";


    delete[] inputArray;

    return 0;
}
```

2)WAP to implement radix sort

```
// radix sort
#include <iostream>
#include <cmath>
using namespace std;
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;
}
void countingSort(int arr[], int n, int exp) {
    int* output = new int[n];
    int* count = new int[10]();

    for (int i = 0; i < n; i++)
        count[(arr[i] / exp) % 10]++;
    for (int i = 1; i < 10; i++)
        count[i] += count[i - 1];
    for (int i = n - 1; i >= 0; i--) {
        output[count[(arr[i] / exp) % 10] - 1] = arr[i];
        count[(arr[i] / exp) % 10]--;
    }
    for (int i = 0; i < n; i++)
        arr[i] = output[i];
```

```
    delete[] output;
    delete[] count;
}
void radixSort(int arr[], int n) {

    int max = getMax(arr, n);
    for (int exp = 1; max / exp > 0; exp *= 10)
        countingSort(arr, n, exp);
}
int main() {
    int N;
    cout << "Enter the number of elements: ";
    cin >> N;

    int* arr = new int[N];
    cout << "Enter the elements: ";
    for (int i = 0; i < N; i++)
        cin >> arr[i];

    radixSort(arr, N);

    cout << "Sorted array: ";
    for (int i = 0; i < N; i++)
        cout << arr[i] << " ";
    delete[] arr;

    return 0;
}
```

3)WAP to implement bucket sort

```
//bucket sort
#include <iostream>
#include <algorithm>
using namespace std;
void bucketSort(int arr[], int n) {

    int maxVal = arr[0];
    for (int i = 1; i < n; i++) {
        if (arr[i] > maxVal)
            maxVal = arr[i];
    }
    int bucketCount = maxVal + 1;
    int** buckets = new int*[bucketCount];
    int* bucketSizes = new int[bucketCount]();
    for (int i = 0; i < bucketCount; i++) {
        buckets[i] = new int[n];
    }
    for (int i = 0; i < n; i++) {
        int bucketIndex = arr[i];
        buckets[bucketIndex][bucketSizes[bucketIndex]++] = arr[i];
    }
    int index = 0;
    for (int i = 0; i < bucketCount; i++) {
        if (bucketSizes[i] > 0) {
            sort(buckets[i], buckets[i] + bucketSizes[i]);
            for (int j = 0; j < bucketSizes[i]; j++) {
                arr[index++] = buckets[i][j];
```

```
            }
        }
    }

    for (int i = 0; i < bucketCount; i++) {
        delete[] buckets[i];
    }
    delete[] buckets;
    delete[] bucketSizes;
}
int main() {
    int N;
    cout << "Enter the number of elements: ";
    cin >> N;
    int* arr = new int[N];

    cout << "Enter the elements: ";
    for (int i = 0; i < N; i++)
        cin >> arr[i];
    bucketSort(arr, N);
    cout << "Sorted array: ";
    for (int i = 0; i < N; i++)
        cout << arr[i] << " ";

    delete[] arr;

    return 0;
}
```