#include<iostream>

using namespace std;

int getMax(int arr[], int n)

{

    int mx = arr[0];

    for (int i = 1; i < n; i++)

        if (arr[i] > mx)

            mx = arr[i];

    return mx;

}

void countSort(int arr[], int n, int exp)

{

    int output[n]; // output array

    int i, count[10] = {0};

    // Store count of occurrences in count[]

    for (i = 0; i < n; i++)

        count[ (arr[i]/exp)%10 ]++;

    // Change count[i] so that count[i] now contains actual

    //  position of this digit in output[]

    for (i = 1; i < 10; i++)

        count[i] += count[i - 1];

    // Build the output array

    for (i = n - 1; i >= 0; i--)

    {

        output[count[ (arr[i]/exp)%10 ] - 1] = arr[i];

        count[ (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] = output[i];

}

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);

}

void print(int arr[], int n)

{

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

}

int main()

{

    int arr[] = {170, 45, 75, 90, 802, 24, 2, 66};

    int n = sizeof(arr)/sizeof(arr[0]);

    radixsort(arr, n);

    print(arr, n);

    return 0;

}