**Monk and Sorting Algorithm**

Monk recently taught Fredo about sorting. Now, he wants to check whether he understood the concept or not. So, he gave him the following algorithm and asked to implement it:

*Assumptions*:  
We consider the rightmost digit of each number to be at index 1, second last at index 2 and so on till the leftmost digit of the number.  
Meaning of ith chunk: This chunk consists of digits from position 5∗i to 1+5∗(i−1) in the given number. If there is no digit at some position in the number, take the digit to be 0.

#include<bits/stdc++.h>

using namespace std;

#include<vector>

#include<iostream>

#define ll long long

#define chunk\_size 100000

void radixSort(vector <ll> ,ll);

void countSort(vector<ll>&,ll);

int main(){

    int n,i;

    cin>>n;

    ll maximum = 0;

    vector<ll>v(n);

    for(i=0;i<n;i++){

        cin>>v[i];

        maximum=max(maximum,v[i]);

    }

    radixSort(v,maximum);

    return 0;

}

void radixSort(vector <ll> v ,ll maximum){

    for(ll it=1;it<=maximum; it\*=chunk\_size){

        countSort(v,it);

    }

    return;

}

void countSort(vector<ll>&v,ll it){

    int i,n=v.size(),freq[chunk\_size]={0};

    vector<ll> temp(n);

    for(i=0;i<n;i++){

        int pos = (v[i]/it)%chunk\_size;

        freq[pos]++;

    }

    for(i=1;i<chunk\_size;i++)

    {

        freq[i]=freq[i]+freq[i-1];

    }

    for(i=n-1;i>=0;i--){

        int pos = (v[i]/it) % chunk\_size;

        temp [freq[pos] -1] =v[i];

        freq[pos]--;

    }

    for(i=0;i<n;i++){

        v[i]=temp[i];

        cout<<v[i]<<" ";

    }

    cout<<endl;

    return;

}