*#include <iostream>*

*#include <cstdlib>*

*#include <queue>*

*#include <vector>*

*#include <string>*

*#include <math.h>*

**using** **namespace** std**;**

**int** numDigits**(int** k**);**

**int** maxElemLength**(const** vector**<int>** **&**v**);** *//Return # of digits of largest int in v*

**int** GetDigit**(int** number**,** **int** k**);** *// Return the kth digit of number*

vector**<** queue**<int>** **>** ItemsToQueues**(const** vector**<int>** **&**L**,** **int** k**);** *// create and return intermediate array of ten queues*

vector**<int>** QueuesToArray**(**vector**<** queue**<int>** **>** **&**QA**);** *// create and return new list from values of intermediate array of queues*

**void** RadixSort**(**vector**<int>** **&**L**,** **int** numDigits**);** *// calls both functions above to sort the list*

**void** PrintVector**(const** vector**<int>** **&**L**);** *// prints contents of vector with no modification*

**int** main**()**

**{**

vector**<int>** L**;**

L**.**push\_back**(380);**

L**.**push\_back**(95);**

L**.**push\_back**(345);**

L**.**push\_back**(382);**

L**.**push\_back**(260);**

L**.**push\_back**(100);**

cout **<<** "\nUnsorted Array: "**;**

PrintVector**(**L**);**

RadixSort**(**L**,** maxElemLength**(**L**));**

cout **<<** "\n\nSorted Array: "**;**

PrintVector**(**L**);**

cout **<<** "\n" **<<** endl**;**

**return** **0;**

**}**

**int** numDigits**(int** k**)**

**{**

**int** power **=** **0;**

**while((**k **-** pow**(10,** power**))** **>** **0)**

power**++;**

**return** power**;**

**}**

**int** maxElemLength**(const** vector**<int>** **&**v**)**

**{**

**int** biggest **=** **0;**

**for** **(int** i**=0;** i **<** v**.**size**();** i**++)**

**{**

**if** **(**v**.**at**(**i**)** **>=** biggest**)**

biggest **=** v**.**at**(**i**);**

**}**

**return** numDigits**(**biggest**);**

**}**

**int** GetDigit**(int** number**,** **int** k**)**

**{**

*//int numDigit = numDigits(number);*

**int** x **=** number **/** **(**pow**(10,** k**));**

**int** y **=** **(**pow**(10,** k**))** **\*** x**;**

**int** z **=** number **-** y**;**

**return** z **/** **(**pow**(10,** k**-1));**

**}**

vector**<** queue**<int>** **>** ItemsToQueues**(const** vector**<int>** **&**L**,** **int** k**)**

**{**

vector**<** queue**<int>** **>** step1**;**

**for** **(int** i **=** **0;** i **<** **10;** i**++)**

**{**

queue**<int>** QA**;**

**for** **(int** j**=0;** j **<** L**.**size**();** j**++)**

**{**

**if(**GetDigit**(**L**.**at**(**j**),** k**)** **==** i**)**

**{**

QA**.**push**(**L**.**at**(**j**));**

**}**

**}**

step1**.**push\_back**(**QA**);**

**}**

**return** step1**;**

**}**

vector**<int>** QueuesToArray**(**vector**<** queue**<int>** **>** **&**QA**)**

**{**

vector**<int>** step2**;**

**for** **(int** i**=0;** i **<** QA**.**size**();** i**++)**

**{**

**while(!**QA**.**at**(**i**).**empty**())**

**{**

step2**.**push\_back**(**QA**.**at**(**i**).**front**());**

QA**.**at**(**i**).**pop**();**

**}**

**}**

**return** step2**;**

**}**

**void** RadixSort**(**vector**<int>** **&**L**,** **int** numDigits**)**

**{**

**for** **(int** i **=** **1;** i **<=** numDigits**;** i**++)**

**{**

vector**<** queue**<int>** **>** step **=** ItemsToQueues**(**L**,** i**);**

L **=** QueuesToArray**(**step**);**

**}**

**}**

**void** PrintVector**(const** vector**<int>** **&**L**)**

**{**

**for** **(int** i**=0;** i **<** L**.**size**();** i**++)**

**{**

cout **<<** L**.**at**(**i**)** **<<** " "**;**

**}**

**}**