#include <iostream>

using namespace std;

#define MAX 100

class SLcell {

public:

int data;

int next;

};

class SLlist {

private:

SLcell r[MAX];

int keynum;

int recnum;

public:

SLlist(int n,int m) {

this->keynum = n;

this->recnum = m;

int a;

for (int i = 1; i <= m; i++) {

cin >> a;

this->r[i].data = a;

}

for (int i = 0; i < this->recnum; ++i)this->r[i].next = i + 1;

this->r[this->recnum].next = 0;

}

void Cout() {

for (int i = 1; i != 0; i = this->r[i].next) {

cout << this->r[i].data <<" ";

}

}

friend void Distribute(SLlist& l, int i, int\*& f, int\*& e);

friend void Collect(SLlist& l, int i, int\*& f, int\*& e);

friend void RadixSort(SLlist& l);

};

int ord(int n, int i) {

for (int j = i; j > 1; j--) {

n = n / 10;

}

return n % 10;

}

void Distribute(SLlist& l, int i, int\*& f, int\*& e) {

for (int j = 0; j < 10; j++)f[j] = 0;

for (int p = l.r[0].next; p; p = l.r[p].next) {

int j = ord(l.r[p].data,i);

if (!f[j])f[j] = p;

else l.r[e[j]].next = p;

e[j] = p;

}

}

void Collect(SLlist& l, int i, int\*& f, int\*& e) {

int j = 0;

for (j = 0; !f[j]; j++);

l.r[0].next = f[j];

int t = e[j];

while (j < 10) {

for (j++; j < 9 && !f[j]; j++);

if (j >= 10)break;

if (f[j]) { l.r[t].next = f[j]; t = e[j]; }

}

l.r[t].next = 0;

}

void RadixSort(SLlist& l) {

int \*f, \*e;

f = new int[10];

e = new int[10];

for (int i = 1; i <= l.keynum; ++i) {

Distribute(l, i, f, e);

Collect(l, i, f, e);

}

delete[]f;

delete[]e;

}

int main()

{

SLlist L(3, 5);

L.Cout();

cout << endl;

RadixSort(L);

L.Cout();

}

