#include <iostream>

using namespace std;

class Sqlist{

private:

int\* r;

int length;

public:

Sqlist(int n) {

this->length = n;

this->r = new int[n+1];

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

int a;

cin >> a;

r[i] = a;

}

}

~Sqlist() {

delete[]this->r;

}

int Getr(int i) {

return this->r[i];

}

friend void HeapAdjust(Sqlist& L, int s, int m);

friend void CreatHeap(Sqlist& L);

friend void HeapSort(Sqlist& L);

};

void HeapAdjust(Sqlist& L, int s, int m) {

int rc = L.r[s];

for (int j = 2 \* s; j <= m; j \*= 2) {

if (j < m && L.r[j] < L.r[j + 1])++j;

if (rc >= L.r[j])break;

L.r[s] = L.r[j]; s = j;

}

L.r[s] = rc;

}

void CreatHeap(Sqlist& L) {

int n = L.length;

for (int i = n / 2; i > 0; i--) {

HeapAdjust(L, i, n);

}

}

void HeapSort(Sqlist& L) {

CreatHeap(L);

for (int i = L.length; i > 1; --i) {

int x = L.r[1];

L.r[1] = L.r[i];

L.r[i] = x;

HeapAdjust(L, 1, i - 1);

}

}

int main()

{

Sqlist L(5);

for (int i = 1; i <= 5; i++) {

cout << L.Getr(i) << " ";

}

cout << endl;

HeapSort(L);

for (int i = 1; i <= 5; i++) {

cout << L.Getr(i)<<" ";

}

}

