#include <iostream>

#include<algorithm>

#include<vector>

using namespace std;

void upheapify(vector<int> &heap,int idx){

if(idx==0){

return;

}

int parent\_idx=(idx-1)/2;

if(heap[parent\_idx]<heap[idx]){

swap(heap[parent\_idx],heap[idx]);

upheapify(heap,parent\_idx);

}

else{

return;

}

}

void insert(vector<int> &heap, int key){

heap.push\_back(key);

upheapify(heap,heap.size()-1);

}

void downheapify(vector<int> &heap,int idx){

int leftidx=2\*idx+1;

int rightidx=2\*idx+2;

if(leftidx>=heap.size() && rightidx>=heap.size()){

return;

}

int largestidx=idx;

if(leftidx<heap.size() && heap[leftidx]>heap[largestidx]){

largestidx=leftidx;

}

if(rightidx<heap.size() && heap[rightidx]>heap[largestidx]){

largestidx=rightidx;

}

if(largestidx==idx)

return;

swap(heap[largestidx],heap[idx]);

downheapify(heap,largestidx);

}

void deletePeek(vector<int> &heap,int k){

int idx=-1;

for(int i=0;i<heap.size();i++){

if(heap[i]==k){

idx=i;

break;

}

}

swap(heap[idx],heap[heap.size()-1]);

heap.pop\_back();

downheapify(heap,idx);

}

void display(vector<int> &heap){

for(int i=0;i<heap.size();i++){

cout<<heap[i]<<" ";

}

cout<<endl;

}

int main() {

vector<int> heap;

int n,x,k;

cin>>n;

for(int i=0;i<n;i++){

cin>>x;

insert(heap,x);

}

display(heap);

cin>>k;

deletePeek(heap,k);

display(heap);

return 0;

}

Input-

11

100 40 43 5 2 9 20 10 20 35 38

40

Output-

100 40 43 20 38 9 20 5 10 2 35

100 38 43 20 35 9 20 5 10 2