#include <vector>

#include <set>

#include <iostream>

using namespace std;

int solution(vector<int> &A, vector<int> &B)

{

int eaten = 0;

std::vector <int>::iterator itw;

std::vector <int>::iterator itwnext;

std::vector <int>::iterator itd;

std::vector <int>::iterator itdnext;

while(1)

{

if(A.size() == 1)

break;

eaten = 0;

itw = A.begin();

itwnext = ++A.begin();

itd = B.begin();

itdnext = ++B.begin();

while(itwnext != A.end())

{

if(\*itd != \*itdnext)

{

if(\*itw > \*itwnext)

{

A.erase(itwnext);

B.erase(itdnext);

}else

{

A.erase(itw);

B.erase(itd);

}

eaten = 1;

continue;

}

itw = itwnext;

itwnext++;

itd = itdnext;

itdnext++;

}

if(eaten == 0)

break;

}

}

/\*There may be different implementations of this program present. This program is memeory efficient and possibly computationally taxing.\*/

int main()

{

//vector <int> weight {7, 9, 3, 4, 10, 1, 2, 5}; /\*Please uncomment the test data pair weight and direction vectors\*/

//vector <int> direction {0, 0, 1, 1, 1, 0, 1, 1};

//vector <int> weight {1, 2};

//vector <int> direction {0, 0};

//vector <int> weight {1};

//vector <int> direction {0};

vector <int> weight {7,9,3,4,10,1,2,5};

vector <int> direction {1,0,1,1,0,1,1,1};

solution(weight, direction);

for(auto const& wgt:weight)

cout<<wgt<<" ";

cout<<endl;

for(auto const& dir:direction)

cout<<dir<<" ";

return 0;

}