'''

Due to side effects of vaccination in Europe, Few people turned as zombies.

And other people fight againest zombies called as warriors to save their lives.

You are given a list of integers strength[], represent the strength of the people.

all the people running in same line with same speed.

If the strength value is negative the person is a zombie running towards left,

If the strength value is positive the person is a warrior running towards right.

and the absolute value represents their strength.

If a zombie and a warrior meet, the person with smallest strength will die.

If both have the same strength, both will die.

Two persons moving in the same direction will never meet.

Your task is to find out the list of strengths of people who will be alive.

Input Format:

-------------

space separated integers, list of strength[].

Output Format:

--------------

Print the list of strengths people who will be alive.

Sample Input-1:

---------------

4 9 -5

Sample Output-1:

----------------

4 9

Explanation:

------------

The 9 and -5 meet resulting in 9. The 4 and 9 never meet.

Sample Input-2:

---------------

-3 -2 2 3

Sample Output-2:

----------------

-3 -2 2 3

Explanation:

------------

The people with the strengths,

-3 and -2 moving left, 3 and 2 moving right. No one will die.

Sample Input-3:

---------------

8 3 -6

Sample Output-3:

----------------

8

Explanation:

------------

The people with the strengths,

3 and -6 meet, 3 will die, -6 is alive.

And 8 and -6 meet, -6 will die, 8 is alive.

'''

l=list(map(int,input().split()))

i=1

j=0

while(i<len(l) and j<len(l)):

if(l[i]<0 and l[j]>0):

if(abs(l[i])>abs(l[j])):

l.pop(j)

elif(abs(l[j])>abs(l[i])):

l.pop(i)

else:

l.pop(i)

l.pop(j)

i=1

j=0

else:

i+=1

j+=1

print(l,end=" ")

There is a set of 'N' bags contains apples.

Each bag carries few number of apples.

You need to pick 'K' bags everytime from left to right.

and find the bag which contains max number of apples 'M'.

You need to return the value fo M each time.

Bags array indicates the apples count in each bag.

NOTE:

-----

Can you solve it in linear time?

Input Format:

-------------

Line-1: Two space separated integers, N and K

Line-2: N space separated integers,

Output Format:

--------------

Print the (N-K+1) integer output.

Sample Input-1:

---------------

8 3

1 2 3 5 4 6 8 7

Sample Output-1:

----------------

3 5 5 6 8 8

Explanation:

------------

K Bags Max

----------- -----

1 2 3 3

2 3 5 5

3 5 4 5

5 4 6 6

4 6 8 8

6 8 7 8

import java.util.\*;

public class Main{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int m=sc.nextInt();

int[] arr=new int[n];

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

arr[i]=sc.nextInt();

}

ArrayList<Integer> l=new ArrayList<>();

for(int i=0;i<n-m+1;i++){

int max=Integer.MIN\_VALUE;

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

if(arr[j]>max){

max=arr[j];

}

}

l.add(max);

}

for(Integer i:l){

System.out.print(i+" ");

}

}

}

A transitive law is "If a == b and b == c, then a == c"

Which also says "If a == b and b != c, then a != c"

You will be given a list of strings relations[], i.e, like p==q or p!=q.

Your task is to find out whether all the relations follow the transitive law or not.

If all of them followed return true, otherwise return false.

Input Format:

-------------

Space separated strnigs, list of relations

Output Format:

--------------

Print a boolean value, whether transitive law is obeyed or not.

Sample Input-1:

---------------

a==b c==d c!=e e==f

Sample Output-1:

----------------

true

Sample Input-2:

---------------

a==b b!=c c==a

Sample Output-2:

----------------

false

Sample Input-3:

---------------

a==b b==c c!=d d!=e f==g g!=d

Sample Output-3:

----------------

True

import java.util.\*;

public class Main{

static int[] parents=new int[26];

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

String s=sc.nextLine();

String[] arr=s.split(" ");

for(int i=0;i<26;i++){

parents[i]=i;

}

for(int i=0;i<arr.length;i++){

if(arr[i].charAt(1)=='='){

union(arr[i].charAt(0)-'a',arr[i].charAt(3)-'a');

}

}

boolean flag=true;

for(int i=0;i<arr.length;i++){

if(arr[i].charAt(1)=='!'){

if(find(arr[i].charAt(0)-'a')==find(arr[i].charAt(3)-'a')){

flag=false;

break;

}

}

}

System.out.println(flag);

}

public static void union(int a,int b){

int aa=find(a);

int bb=find(b);

if(aa!=bb){

parents[aa]=bb;

}

}

public static int find(int a){

if(parents[a]==a){

return a;

}

return find(parents[a]);

}

}

In an International school 9th standard students have been assigned a task.

Given a list of numbers nums[] and three values p, q and r.

They have to solve the equation function(x)=p(x^2)+q(x)+r,

for each number(x) in the given list nums[],

and print the resultant values in the sorted order.

Input Format:

-------------

Line-1: An integer N, size of list nums[].

Next N lines: space separated integers, nums[]

Last Line: 3 space separated integers, P, Q and R.

Output Format:

--------------

Print a list of numbers[].

Sample Input-1:

---------------

4

-3 -2 2 4

2 3 5

Sample Output-1:

----------------

7 14 19 49

Sample Input-2:

---------------

4

-3 -2 1 2

-2 3 5

Sample Output-2:

----------------

-22 -9 3 6

Only 87.5 percent passed

import java.util.\*;

public class Main{

static int p,q,r;

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int n=sc.nextInt();

int[] arr=new int[n];

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

arr[i]=sc.nextInt();

}

p=sc.nextInt();

q=sc.nextInt();

r=sc.nextInt();

int i=0;

int j=n-1;

int[] ans=new int[n];

int num=n-1;;

while (i<=j){

if(quad(arr[i])>quad(arr[j])){

ans[num]=quad(arr[i]);

i++;

// num-=1;

}

else{

ans[num]=quad(arr[j]);

j--;

// num-=1;

}

num--;

}

System.out.println(Arrays.toString(ans));

}

public static int quad(int x){

return p\*x\*x+q\*x+r;

}

}