**Q.1 >** Recently Oz has found a magical string consisting of single digit "1". After experimenting on the string, Oz found a weird magical property of the string that is whenever he touches the string then each digit "1" of string changed to digit "0" and each digit "0" of string changed to "01". Oz found this property interesting and immediately asked a question to RK : "How many 1's and 0's will be in the magical string if he touches the string M times ?"

Input :

The first line contains the number of test cases T . Each test case consists of a positive integer - M .

Output :

For each test case output two space-separated integers, number of 1's and number of 0's in the magical string if Oz touches the string M times.

Constraints :

1<= T <=20

1<= M <=90

SAMPLE INPUT

2

1

2

SAMPLE OUTPUT

0 1

1 1

**Q.2>** """

Given an array A of size N. Given Q operations, each operation contains an integer D. In each operation you have to divide all the elements of the array by D.

For example, for each operation with a given D, the new array A would be:

A[0] / D, A[1] / D, A[2] / D, ..... , A[N-1] / D

Finally, after processing all the operations you have to print the final array after Q operations.

Note : The result of the each division will be an integer, for example 5 / 2 = 2

Input :

First line of input contains a single integer N denoting number of elements in the array A. Next line of input contains N space separated integers denoting the elements of array A. Next line contains Q denoting number of operations. Next Q lines contains a single integer D by which divide the elements of the array.

Output :

Print single line containing N space separated integers after processing Q operations.

INPUT VALUE

5

50 20 18 27 19

3

2

3

2

Constraints:

1<=N<=100000

1<=Ai<=1000000

1<=Q<=100000

1<=D<=1000

In operation 1 after dividing the whole array by D=2, the resultant array will be : [25, 10, 9, 13, 9] In operation 2 after dividing the array from operation 1 by 3, the resultant array will be : [8, 3, 3, 4, 3] In operation 3 after dividing the array from operation 2 by 2, the resultant array will be : [4, 1, 1, 2, 1] So, the resultant array will be [4, 1, 1, 2, 1]

**Q.3>** Given a string consisting of opening and closing parenthesis, find length of the longest valid parenthesis substring.

Examples:

Input : ((()

Output : 2

Explanation : ()

Input: )()())

Output : 4

Explanation: ()()

Input: ()(()))))

Output: 6

Explanation: ()(())

Input:

**Q.4>**  kidnapper wrote a ransom note but is worried it will be traced back to him. He found a magazine and wants to know if he can cut out whole words from it and use them to create an untraceable replica of his ransom note. The words in his note are *case-sensitive* and he *must* use whole words available in the magazine, meaning he *cannot* use substrings or concatenation to create the words he needs.

Given the words in the magazine and the words in the ransom note, print Yes if he can replicate his ransom note *exactly* using whole words from the magazine; otherwise, print No.

**Input Format**

The first line contains two space-separated integers describing the respective values of  (the number of words in the magazine) and  (the number of words in the ransom note).   
The second line contains  space-separated strings denoting the words present in the magazine.   
The third line contains  space-separated strings denoting the words present in the ransom note.

**Constraints**

* .
* Each word consists of English alphabetic letters (i.e.,  to  and  to ).
* The words in the note and magazine are *case-sensitive*.

**Output Format**

Print Yes if he can use the magazine to create an untraceable replica of his ransom note; otherwise, print No.

**Sample Input 0**

6 4

give me one grand today night

give one grand today

**Sample Output 0**

Yes

**Sample Input 1**

6 5

two times three is not four

two times two is four

**Sample output 2**

NO

Q.5> A *left rotation* operation on an array of size  shifts each of the array's elements  unit to the left. For example, if left rotations are performed on array , then the array would become .

Given an array of  integers and a number, , perform  left rotations on the array. Then print the updated array as a single line of space-separated integers.

**Input Format**

The first line contains two space-separated integers denoting the respective values of  (the number of integers) and  (the number of left rotations you must perform).   
The second line contains  space-separated integers describing the respective elements of the array's initial state.

**Constraints**

**Output Format**

Print a single line of  space-separated integers denoting the final state of the array after performing  left rotations.

**Sample Input**

5 4

1 2 3 4 5