The countess' pearl necklace is known for the quality and quantity of its pearls. She wears it at big parties, giving it three or four laps around her neck. The center pearl is the largest, and as we move towards the ends they are strictly smaller. The pearls are perfectly selected and placed, what makes the necklace fully symmetrical with respect to the central pearl.

During the last dance, the closure broke and the pearls rolled on the floor. The guests tried to retrieve them all by stranding them on a new string as they were found. The next day, the countess called her jeweler to put them back in the right order. The jeweler measures the diameter of each pearl and proceeds to remake the collar placing first the central pearl and then the following ones in size until arriving at the ends. The countess does not care if a pearl has been lost, as long as the necklace remains completely symmetrical.

## Input

Each line of the entry forms a test case, consisting of a list of positive numbers separated by spaces. Each one represents the diameter of one of the pearls in the order in which they were collected by the guests. The perfection of the necklace is legendary, so the diameter is measured with a unit that many consider infinitesimal, although the diameter will always be less than $2^{31}$. Each list has $0 < pearls < 1{,}000$ and finishes always with a zero.

The last test case, which should not be processed, contains an empty collar, represented by a list with a single zero.

## Output

For each test case, the program writes '`NO`' if it is not possible to form a symmetrical collar with each and every one of the pearls found, so that the larger pearl is in the center. Otherwise, the diameters of the pearls will be written, separated by spaces, as they have been arranged in the collar, from one end to the other.

## Sample Input

```
2 3 2 3 5 7 5 0
2 2 7 10 0
10 0
0
```

## Sample Output

```
2 3 5 7 5 3 2
NO
10
```