

“Commander! Commander! Please wake up commander!”

“... mmmph. What time is it?”

“4:07 am, Commander. The following message just arrived on the emergency zeta priority classified scrambler, marked your eyes only.”

You grudgingly take the letter, rub the sleep from your eyes, fleetingly wish that the 'Backer closed at an earlier hour, and start to read.

``Dear StarWars SDI Commander,

Bad news, buddy. Crazy Boris had a bit too much vodka last night and when he woke up this morning, instead of the snooze button on his alarm clock, he ... well, let me put it this way: we've got tons of nuclear missiles flying this way. Unfortunately, all that we have is a chart of the altitudes at which the missiles are flying, arranged by the order of arrivals. Go for it, buddy. Good luck.

Secretary of Defense

P.S. Hilly and Bill say hi.``

To make things worse, you remember that SDI has a fatal flaw due to the budget cuts. When SDI sends out missiles to intercept the targets, every missile has to fly higher than the previous one. In other words, once you hit a target, the next target can only be among the ones that are flying at higher altitudes than the one you just hit.

For example, if the missiles are flying toward you at heights of 1, 6, 2, 3, and 5 (arriving in that order), you can try to intercept the first two, but then you won't be able to get the ones flying at 2, 3, 5 because they are lower than 6. Your job is to hit as many targets as possible. So you have to quickly write a program to find the best sequence of targets that the flawed SDI program is going to destroy.

Russian war tactics are fairly strange; their generals are sticklers for mathematical precision. Their missiles will always be fired in a sequence such that there will only be *one* solution to the problem posed above.

## Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The input to your program will consist of a sequence of integer altitudes, each on a separate line.

## Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

Output from your program should contain the total number of targets you can hit, followed by the altitudes of those targets, one per line, in the order of their arrivals.

## Sample Input

```
1
1
6
2
3
5
```

## Sample Output

```
Max hits: 4
1
2
3
5
```