SPOJ Problem Set

383. European railroad tracks

Problem code: EUROPEAN

As you may already know, different countries in Europe use different railroad systems. Not only do they use different voltages for their trains, but also the distance between the two rails (gauge) differs. The following table shows some railway gauges used:

Broad gauge (Spain):	**1674 mm**
Broad gauge (Portugal):	**1665 mm**
Broad gauge (Ireland):	**1600 mm**
Broad gauge (Finland):	**1524 mm**
Broad gauge (former USSR):	**1520 mm**
Standard gauge:	**1435 mm**
Narrow gauge (meter gauge):	**1000 mm**

A museum has trains from several countries. It needs tracks for every train type in order to show visitors the trains in use. However, since only one train is used at a time, a rail can be used by trains of different types. It follows that for *n* trains, each requiring a different railway gauge, *n + 1* rails are sufficient (each train uses the leftmost rail and a rail that has exactly the required distance to it). But sometimes it is possible to save even more rails.

Given the required railway gauges, your task is to construct a railway track that can be used by every train and requires the least number of rails. Note that a train can use any two rails, provided the distance between them is right.

Input Specification

The first line of the input file contains a number representing the number of test cases to follow. Each test case starts with an integer *n* (the number of different railway gauges required). The next line contains *n* integers between *1000* and *5000*, each defining one required railway gauge.
You can assume that $1 \le n \le 8$. Moreover, for every test case in the input file, there will be a solution requiring at most *5* rails.

Output Specification

The output for each test case consists of three lines:
The first line is of the form "Scenario #X", where X is the test case number starting with 1. The second line describes the solution your program has found; first your program should print how many rails are needed, followed by a colon, then the positions of each rail in increasing order (the first rail should be at position 0). The third line should be blank. If there are several solutions with the minimum number of rails, any one will do.

Sample Input

```
3
4
1524 1520 1609 1435
3
1000 1520 1600
6
1000 2000 3000 4000 1500 2500
```

Sample Output

```
Scenario #1
4: 0 1520 1609 3044

Scenario #2
4: 0 1000 1520 1600

Scenario #3
5: 0 1500 3000 4000 5000
```

Added by:	Adrian Kuegel
Date:	2005-06-24
Time limit [s]:	30
Source limit [B]:	50000
Resource:	University of Ulm Local Contest 2005